Acceptable Costs of Minimax Regret Equilibrium: A Solution to Security Games with Surveillance-Driven Probabilistic Information☆

Wenjun Ma*[a], Kevin McAreavey[b], Weiru Liu[b], Xudong Luo[c]

[a]*School of Computer Science, South China Normal University, Guangzhou, China*
[b]*School of Computer Science, Electrical and Electronic Engineering, and Engineering Maths, University of Bristol, UK*
[c]*Department of Information and Management Science, Guangxi Normal University, Guilin, China.*

Abstract

We extend the application of security games from offline patrol scheduling to online surveillance-driven resource allocation. An important characteristic of this new domain is that attackers are unable to observe or reliably predict defenders' strategies. To this end, in this paper we introduce a new solution concept, called *acceptable costs of minimax regret equilibrium*, which is independent of attackers' knowledge of defenders. Instead, we study how a player's decision making can be influenced by the emotion of regret and their attitude towards loss, formalized by the *principle of acceptable costs of minimax regret*. We then analyse properties of our solution concept and propose a linear programming formulation. Finally, we prove that our solution concept is robust with respect to small changes in a player's degree of loss tolerance by a theoretical evaluation and demonstrate its viability for online resource allocation through an experimental evaluation.

Keywords: security game, real-time resource allocation, minimax regret, loss aversion, intelligence surveillance system, decision support

☆This is a significant extension of its conference version in (Ma et al., 2015).

*Corresponding author.

URL: phoenixsam@sina.com (Wenjun Ma*), kevin.mcareavey@gmail.com (Kevin McAreavey), weiru.liu@bristol.ac.uk (Weiru Liu), luoxd@mailbox.gxnu.edu.cn (Xudong Luo)

1. Introduction

The problem of how to best allocate limited security resources for protecting critical infrastructure and the general public has attracted much interest in recent years (Tambe, 2011; Kar et al., 2016; Lou et al., 2017). Most of these studies address the problem of security patrol scheduling. At the same time, the development of intelligent surveillance systems has progressed to the extent that threats can be detected in real-time and analytic information can be used to assess the intentions of an attacker, *e.g.*, using event inference (Wasserkrug et al., 2008; Ma et al., 2010, 2014; Hong et al., 2016). For this reason, there is a clear opportunity to understand how other types of security resource allocation can be informed by real-time intelligent surveillance systems.

In fact, we can distinguish two sorts of security resource allocation problems within intelligent surveillance systems. (i) Offline patrol scheduling for security teams, security checkpoints, and the deployment of surveillance devices, such as CCTV (Closed Circuit Television) cameras or other sensors-networks. Given that attackers can learn such offline methods by observations, this sort of security resource allocation problems can be solved by methods based on Bayesian Stackelberg games (Tambe, 2011). In this paper, we do not consider such problems. (ii) Online attacker threat prevention. For this sort of security resource allocation problems, firstly a defender will take the action after the surveillance system has detection potential of the attackers and predicted their intention. This means that it is impossible for the attackers to observe the defender's actual strategy execution in real time and then change their strategy accordingly. Secondly, the environment is highly dynamic and the defender will execute relevant strategies infrequently, so it is difficult for the attackers to learn the defender's behavior from historical data. In this situation, the attackers havep very limited knowledge about the possible action of the defender. This is the problem that we will focus in this paper, which we refer to such as a surveillance-driven security resource allocation (SDSRA) problem. For example, let us consider the following scenario:

Example 1. *There are three assets in a high street shopping area: a shopping mall, a foreign currency exchange shop, and a hotel. An intelligent surveillance system has detected that a person has been excessively loitering in the area. A combination of event reasoning and a comparison of facial image capture with historical data suggests that the person may be a terrorist or an armed robber or a pickpocket. Suppose there is only one security team available—which asset should they be assigned to protect?*

In the literature, security resource allocation is commonly addressed using a special type of (non-zero-sum) Bayesian Stackelberg game known as a security game (Pita et al., 2009; Tambe, 2011; Ma et al., 2013a). As standard Bayesian Stackelberg games, a security game is a two player game where a defender (leader) commits to a strategy, then an

attacker (follower) chooses their best response with knowledge of the defender's commitment. And in this kind of security game, uncertainty over the preferences of the attacker is modelled as a probability distribution over possible attacker types, (*i.e.*, different types mean different attack preferences). However, in contrast to the standard Bayesian Stackelberg model, security games have the assumption that what is good for the attacker is bad for the defender and vice versa. By definition, the goal is to mitigate the effects of the attacker, while dealing with uncertainty over their attack preferences, by choosing a strategy for the defender which will influence the attacker's choice to the benefit of the defender. This is formalized in game theory by a solution concept. Many solution concepts have been proposed to handle different real-world challenges in security games, including the Bayes-Nash Equilibrium (BNE) (Korzhyk et al., 2011), the Strong Stackelberg Equilibrium (SSE) (Tambe, 2011), robust non-equilibrium solutions against humans (Pita et al., 2010) and worst-case approaches for interval uncertainties (Kiekintveld and Kreinovich, 2012). The vast majority employ the SSE.

A key characteristic of the SSE is that it assumes the attacker has perfect knowledge of the defender's strategy, whether by direct observation of the strategy execution or by learning from historical information. Alternative solution concepts make similarly strong assumptions, *e.g.*, the BNE assumes the attacker has perfect knowledge of the defender's utility[1] values and the defender's assessment of the attacker type. However, more and more researchers realise that these assumptions are unrealistic in the context of many real-world applications, and there has been some work towards their relaxation (Balcan et al., 2015; Fang et al., 2015; Kar et al., 2015; Sinha et al., 2016). As we will see in more detail in Section 2, these existing methods are not well suitable to solve SDSRA problems. This is because in SDSRA, the attacker is unable to observe the defender's strategy execution, while their knowledge of the defender is often too limited to make any meaningful predictions. In fact, when the attacker is faced with incomplete information about the defender, predictions of the attacker's strategy in existing models are inconsistent with empirical observations (Crawford et al., 2013; Goeree and Holt, 2001). Given that the defender's strategy is a best response to their prediction of the attacker's strategy, which in turn is based on the attacker's knowledge of the defender, then this may result in unacceptable losses for the defender if their prediction of the attacker's strategy is incorrect. In other words, in SDSRA, dependence on the attacker's knowledge of the defender—even if the imperfection of this knowledge is accepted—may result in serious negative consequences for the defender. Clearly, an approach which is independent of the attacker's knowledge of the defender is robust to this form of uncertainty. On the other hand, while existing robust models (*e.g.*, (Kiekintveld and Kreinovich, 2012; Nguyen et al., 2014a; Pita et al., 2010))

[1] In this paper, the terms *utility* and *payoff* are synonymous.

are based on similar assumptions, they often result in overly conservative behaviour, (*i.e.*, they select a strategy for the defender which results in a lower expected utility than necessary).

To address this issue, in this paper we will propose a new solution concept, called *acceptable costs of minimax regret equilibrium*, to security games in the SDSRA domain based on the *principle of acceptable costs of minimax regret* from decision theory. In particular, our solution concept is based upon the concepts of loss-aversion and regret (Ma et al., 2017) in order to predict the behaviour of the attacker, independent of their knowledge of the defender. Then defender uses this prediction to select a strategy. More specifically, our solution concept has three assumptions: **(A1)** each player is loss-averse; **(A2)** each player seeks to minimize their maximum regret with respect to their degree of loss-aversion; and **(A3)** the defender can be aware of the attacker's utility values but the attacker cannot be aware of the defender's utility values.

In this paper, our main contributions are as follows. (i) We extend the application of security games to the surveillance-driven security resource allocation problem. (ii) We propose a new solution concept, which balances the defender's desire for rewards from successful threat prevention with their aversion to losses. (iii) We analyze some properties of this new solution concept. (iv) We propose a linear programming formulation. (v) We prove the robustness of our results. And (vi) we demonstrate its viability for real-world SDSRA problems with an experimental evaluation.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 formally defines the game-theoretic model of an SDSRA problem. Section 3 recaps some basic solution concepts in game theory based on our definition of an SDSRA problem. Section 5 explains the three assumptions underpinning our new solution concept. Section 6 proposes our new solution concept and analyzes its properties. Section 7 demonstrates our solution concept with a detailed scenario, proves the robustness of our results, and provides an experimental evaluation. Finally, Section 8 concludes the paper with possibilities for future work.

2. Related Work

In recent years, there has been an increase in the deployment of intelligent surveillance systems, largely in response to the high demand for identifying and preventing threats to public safety, *e.g.*, suspect object tracking (Du et al., 2018) and anti-social behavior analysis (Zhang et al., 2016; Ma et al., 2010). The advances in game-theoretic solutions to security raise a number of interesting research questions. Although much work has addressed the issue of uncertainty over attacker types (Kiekintveld et al., 2013; Ma et al., 2013b; Nguyen et al., 2014b; Yang et al., 2012; Clempner, 2017), to the best of our knowl-

edge, no work of this kind has studied how to employ these solutions in intelligent surveillance systems for reactive, online security resource allocation. In the literature, some non-equilibrium solution concepts have been proposed to handle uncertainty in security games as well as to address the issue of human adversaries with bounded rationality, bounded memory and incomplete knowledge of the defender's strategy. Generally, we can divide these non-equilibrium solution concepts into two categories:

Robust solutions: This sort of solution concept focuses on how to guarantee the defender's expected utility in the worst case, based on an imperfect prediction of the attacker's behavior or on other aspects of uncertainty in security games. Some examples are robust non-equilibrium solutions against humans (Pita et al., 2010), worst-case approaches (Kiekintveld and Kreinovich, 2012), the Interval Security Game (ISG) model (Kiekintveld et al., 2013), the minimax regret method for interval uncertainty (Nguyen et al., 2014b), the monotonic maximin method for an attacker's bounded rationality (Jiang et al., 2013), the MATCH algorithm (Pita et al., 2012), and unified robust algorithms (Nguyen et al., 2014a).

Utility maximizing solutions: This sort of solution concept focuses on how to maximize the defender's expected utility, based on an appropriate prediction of the attacker's behavior obtained by human decision-making models or by learning algorithms. Some examples are the quantal response equilibrium against human adversaries with bounded rationality (Yang et al., 2012) and the subjective utility quantal response method which integrates human decision-making models into game-theoretic algorithms (Nguyen et al., 2013). At the same time, inspired by developments in machine learning, some have used learning algorithms to determine the optimal strategy of the defender by learning how the attacker is likely to respond (Balcan et al., 2015; Blum et al., 2014; Kar et al., 2015). For example Fang et al. (2015) propose a planning algorithms and a learning framework to address a problem in the green security domain in which adversaries lack the resources to fully observe the defender's strategy. Also, Sinha et al. (2016) propose the Probably Approximately Correct model to learn the adversary response function in Stackelberg security games.

In most cases, these non-equilibrium solution concepts are based on the Stackelberg game framework. Thus, most of them require that the attacker has at least partial knowledge of the defender's strategy commitment. For example, the logit quantal response function adopted in (Yang et al., 2012; Jiang et al., 2013; Nguyen et al., 2013; Fang et al., 2015; Sinha et al., 2016) is a function $P : \mathbb{R}^n \rightarrow Y$ from the vector of expected utilities for an attacker's strategy to a probability distribution over the strategy. The function is defined as: $P_j(\overrightarrow{u}) = \frac{e^{\lambda u_j}}{\sum_{j'} e^{\lambda u_{j'}}}$, where $P_j(\overrightarrow{u})$ is the probability of an attacker playing strategy j given

the expected utility vector $\overrightarrow{u} \in \mathbb{R}^n$ and an error parameter $\lambda \geq 0$. Importantly, regardless of whether vector $\overrightarrow{u}$ is obtained by an expected utility function or by a subjective utility function, knowledge of the defender's coverage probability for each target is required to construct the expected utility vector. In other words, it requires at least partial knowledge of the defender's strategy selection. There are two exceptions which can model the situation in which the attacker has no knowledge of the defender's strategy: in some worst case approaches (Pita et al., 2010; Kiekintveld and Kreinovich, 2012; Nguyen et al., 2014a) and in some quantal response approaches (Yang et al., 2012; Nguyen et al., 2013; Jiang et al., 2013; Blum et al., 2014; Balcan et al., 2015; Kar et al., 2015). However, there are important limitations with each. For the worst case solutions, in a sense, the defender disregards any knowledge that they may have about the attacker's utilities, so it is arguable that these approaches are overly conservative (Jiang et al., 2013). That is, they select a strategy for the defender which results in a lower expected utility than is necessary. For the quantal response solutions, the attacker's ignorance over the defender's strategy is modelled by a mixed strategy which assigns equal probabilities to all pure strategies, (*i.e.*, by applying the principle of indifference). This raises two questions. Firstly, is it acceptable to apply the principle of indifference to describe the attacker's complete ignorance? Secondly, is it reasonable to assume that the attacker still pursues a higher expected utility when they have no idea about the defender's strategy? In the literature, many psychological experiments and analysis of economics have provided strong evidence to support that applying the principle of indifference in this way can produce results that do not correspond to actual human decision making (Savage, 1951; Ellsberg, 1961; Kahneman and Tversky, 1979; Kahneman, 2003).

In the case of utility maximizing solutions, these methods often explore a new direction in security games where machine learning is used to learn and predict adversary behavior using available data about player interactions. However, these data-driven methods all require significant amounts of historical data about player interactions in order to learn a reasonably representative model of adversary behavior. Clearly, as argued by Fang et al. (2015), Sinha et al. (2016), and Tambe (2011), there is very limited data available in infrastructure security domains (which we focus on in SDSRA problems) in comparison to green security domains. Thus, it is unrealistic to assume that we can learn an adequate model of player interactions from historical data. As mentioned previously, in SDSRA problems, the assumption of complete or partial knowledge of the defender's strategy commitment is unrealistic for two reasons. Firstly, the defender selects a strategy in response to the detection of an attacker, which means that it is impossible for the attacker to observe the defender's actual strategy execution in real-time and then update change their own strategy accordingly. Secondly, the environment is highly dynamic and the defender will execute relevant strategies infrequently, so it is difficult for the attacker to learn

the defender's likely behavior from historical data. Due to these reasons plus the limitations of the worst cases approaches and the principle of indifference, none of the existing concepts of solution to security games are directly applicable to SDSRA problems.

3. Security Games for SDSRA

Generally speaking, an SDSRA problem involves an intelligent surveillance system (*e.g.*, based on CCTV or sensor-networks), which continuously detects potential threats by considering the characteristics and behavior of potential attackers. In real-world SDSRA problems, an attacker may select multiple targets to attack but a defender only has limited resources for protecting these targets. As a result, in order to best allocate the security resources, the defender must online determine which target the attacker is most likely to attack. This contrasts with offline patrol scheduling problems in traditional security games, where the defender attempts to assign all security resources so as to prevent possible attacks. In an SDSRA problem, the defender may instead prefer to retain some security resources so as to protect against the possibility of more significant attacks in the future. Thus, it is reasonable to assume that the defender will not allocate more security resources than the number of attackers. Of course, it is also possible for a single security resource to defend against more than one attacker in real-world applications. This would suggest that there are more attackers than the number of security resources.

However, by our acceptable costs of minimax regret equilibrium, the attackers' strategies are determined by the amounts of security resources and the defender's optimal strategy is determined by the prediction of the attackers' strategies. (More details can read Sections 5 and 6). As a result, after the amounts of security resources is given, the number of attacker will only influence the structure of the utility function (more details about the construction of the utility function will be shown in Definition 1 and the two paragraphs after it). Thus, since the attackers' strategies are determined by the amounts of security resources, if we can understand the attackers' strategy in the case where the amount of security resources is the same as the numbers of attackers, then it is easy to extend our method to predict the attackers' strategy when the amounts of security resources is less than the number of attackers. Hence, after we obtain the attackers' strategies, we can obtain the defender's optimal strategy based on such attackers' prediction strategy easily. In addition, in case of multiple attackers, we assume that the attackers act independently (*e.g.*, they cannot negotiate, cooperate, or share the profit obtained from the game), which is the same assumption made by traditional security games (Pita et al., 2009; Tambe, 2011). For these reasons, and in order to focus on the essence of SDSRA problem, we impose the following assumption about SDSRA problems:

In a given situation, the defender will allocate the same number of security

Target	Covered		Uncovered	
	D	A	D	A
SM	7	-7	-8	9
FCE	4	-2	-3	3
H	6	-6	-6	5

(a) $t_1 = 0.2$.

Target	Covered		Uncovered	
	D	A	D	A
SM	5	-5	-4	5
FCE	7	-8	-6	7
H	6	-5	-5	5

(b) $t_2 = 0.3$.

Target	Covered		Uncovered	
	D	A	D	A
SM	5	-8	-5	7
FCE	5	-7	-4	6
H	7	-7	-6	6

(c) $t_3 = 0.5$.

Table 1: Security game where $\sigma_1 = 0.5$, D stands for Defender and A stands for Attacker.

resources as number of attackers, while the attackers operate independently.

Thus, we can formally define a security game for SDSRA as follows:

Definition 1. *A security game for a Surveillance-Driven Security Resource Allocation (SDSRA) problem is an 8-tuple* $(N, l, T, \mathsf{p}, K, A, \Psi, U)$ *where:*

(i) $N = \{1, 2\}$ *is the set of players, where player* 1 *represents the defender and player* 2 *represents the attacker;*

(ii) l *is the number of attacker that the defender has to face in a scenario;*

(iii) $T = \{t_1, \ldots, t_n\}$ *is a non-empty finite set of* n *possible attacker types;*

(iv) p *is a probability distribution over* T*;*

(v) $K = \{k_1, \ldots, k_m\}$ *is a set of targets;*

(vi) $A = \{A_1, A_2\}$ *where* A_1 *denotes the set of pure strategies (action-plans) for the defender (*i.e.*, player* 1*) and* A_2 *denotes the set of pure strategies for the attacker (*i.e.*, player* 2*) such that there exists a bijection* $f : A_i \to K^l$ *where* l *is the number of attackers;*

(vii) $\Psi = A_1 \times A_2$ *is the set of pure strategy profiles; and*

(viii) $U = \{\mathsf{u}_i^t \mid i \in N, t \in T\}$ *where* $\mathsf{u}_i^t : \Psi \to \mathbb{R}$ *is a utility function for player* i *with respect to attacker type* t*.*

In fact, in an SDSRA security game, in order to assign the utility values to each target for both players, the defender needs to consider the importance of each target for the given player and the effect of intelligent surveillance systems on the attackers (i.e, the patrol scheduling of security team, the checkpoint setting, and the CCTV or sensor-networks

	SM	FCE	H
SM	7, -7	-3, 3	-6, 5
FCE	-8, 9	4, -2	-6, 5
H	-8, 9	-3, 3	6, -6

(a) $t_1 = 0.2$.

	SM	FCE	H
SM	5, -5	-6, 7	-5, 5
FCE	-4, 5	7, -8	-5, 5
H	-4, 5	-6, 7	6, -5

(b) $t_2 = 0.3$.

	SM	FCE	H
SM	5, -8	-4, 6	-6, 6
FCE	-5, 7	5, -7	-6, 6
H	-5, 7	-4, 6	7, -7

(c) $t_3 = 0.5$p.

Table 2: Complete utility matrices for security game where $\sigma_1 = 0.5$.

deployment around each target). For example, although the importance of a control center in an airport is very high, after considering a checkpoint before the entrance of the control center and the sensor-network setting in it, the security team should think the chance of an attacker choose the control center to attack is low. Thus, the defender should reduce the utility value of the control center. In this paper, we assume the utility values are a set of point values that are given by the security team.

Moreover, given that covering or attacking a target with one resource is essentially the same as covering or attacking it with any positive number of resources, the set of pure strategies A_i of a player $i \in N$ can also be represented by a set $D \subseteq \{0,1\}^{|K|}$, where each element $\mathbf{d}_i \in D$ is a coverage vector $\mathbf{d}_i = (d_i^1, \ldots, d_i^{|K|})$ with $\sum_{j=1}^{|K|} d_i^j = l$ such that l is the number of attackers in the game. Thus, each d_i^j says whether target $k_j \in K$ is covered or attacked. Hence, a mixed (randomized) strategy Δ_i for player i is a probability distribution over the set of pure strategies A_i. For this reason, a mixed strategy Δ_i is equivalent to a pure strategy $a_i \in A_i$ when $\Delta_i(a_i) = 1$ and vice versa. When this is the case, we will simply write a_i rather than Δ_i. Also, we may abuse notation and apply a utility function to a mixed strategy profile, (*i.e.*, where one or both players commit to a mixed strategy). Given a mixed strategy profile (Δ_1, Δ_2) and an attacker type t, the utility of player i is defined as follows:

$$\mathsf{u}_i^t(\Delta_1, \Delta_2) = \sum_{a_1 \in A_1} \sum_{a_2 \in A_2} \Delta_1(a_1)\Delta_2(a_2)\mathsf{u}_i^t(a_1, a_2). \tag{1}$$

These types of probability-weighted utilities are referred to as *expected utilities*. Importantly, the "probabilistic surveillance information" that we have referred to is modelled by the probability distribution p over the set of attacker types T.

In Definition 1, we define an SDSRA security game with a utility structure corresponding to the real-world problem of infrastructure security as in Stackelberg security games (Tambe, 2011). As in Stackelberg security games, a simplification in SDSRA security games is to only assign utility values based on whether an asset is *covered* or *uncovered* (refer to Table 1 for an example). In the former, the attacker is unsuccessful (*i.e.*, the de-

fender wins) if they target an asset which is protected by the defender, while in the latter, the attacker is successful (*i.e.*, the attacker wins) if they target an asset but the defender is protecting another. For this reason, we often refer to these assets as *targets* for the attacker and represent them as the set K in Definition 1. In addition, we assume that each player will have positive utility values for pure strategy profiles in which they win, and negative utility values for pure strategy profiles in which they lose. This is a common restriction in the literature which differentiates security games from standard Bayesian Stackelberg games. Moreover, it means that an SDSRA security game satisfies the property that there exists no *dominant strategy* for either player. Specifically, a dominant strategy is a strategy where the utility for the player is always higher than for another strategy, regardless of how the player's opponent might play. Finally, given a mixed strategy Δ_1 for the defender (which specifies the probability of playing each pure strategy), the probability of covering a target k_j is $\sum_{\mathbf{d}_1 \in D} d_1^j \Delta_1(a_j)$. For example, suppose the defender has two coverage vectors: $\mathbf{d}_1 = (0, 1, 1)$ and $\mathbf{d}_1' = (1, 0, 1)$. For the mixed strategy $\Delta_1 = (0.3, 0.7)$, the corresponding vector of coverage probabilities is $0.3 \cdot \mathbf{d}_1 + 0.7 \cdot \mathbf{d}_2 = (0.7, 0.3, 1)$.

Now, we consider the scenario from Example 1 to explain the concepts in Definition 1 as follows:

Example 2 (Example 1 continued)**.** *Suppose in an airport, there are three security targets: a shopping mall (SM), a foreign currency exchange shop (FCE), and a hotel (H). There are three possible types of attacker: a terrorist (t_1), an armed robber (t_2), and a pickpocket (t_3). The scenario can be modelled as an SDSRA security game by Definition 1 as follows:*

(i) It is a security game for SDSRA with two players where player 1 *is the defender and player* 2 *is the attacker. Thus, we have a player set $N = \{1, 2\}$;.*

(ii) The number of attacker is $l = 1$, since the scenario only has one attacker.

(iii) The possible attacker types set is $T = \{t_1, t_2, t_3\}$, where t_1 means he is a terrorist, t_2 means he is an armed robber, and t_3 means he is a pickpocket.

(iv) When the threat is detected, an assessment is made of the possible type of attacker in the form of a probability distribution over the set of the types: $\mathsf{p}(t_1) = 0.2$, $\mathsf{p}(t_2) = 0.3$, and $\mathsf{p}(t_3) = 0.5$.

(v) The targets set is $K = \{SM, FCE, H\}$.

(vi) The pure strategies of each player is described in Table 2: the defender can choose one of these three targets to cover, and the attacker can choose one of these three target to attack. Moreover, we have the pure strategy set for each player that $A_1 =$

	X	Y
A	1, 0	0, 1
B	0, 3	1, 0

Table 3: Aumann and Maschler Game.

$A_2 = \{SM, FCE, H\}$. *Here, an element* $x \in A_1$ *means targets* x *is covered by the defender; and an element* $y \in A_2$ *means target* y *is attacked by the attacker. Also, any pure strategy can describe by a coverage vector. For example, the defender covers SM can be described as* $\boldsymbol{d}_1 = (1, 0, 0)$. *And this method is helpful when the number of resource or attacker is more than 1. For example, a pure strategy describes that the attacker will attack SM and H can be represented as* $\boldsymbol{d}_2 = (1, 0, 1)$.

(vii) The set of pure strategy profiles $\Psi = A_1 \times A_2$. *For example, a pure strategy profile* (FCE, SM) *means that the defender chooses FCE to cover and the attacker chooses SM to attack.*

(viii) The utility matrix for each player with respect to each attacker type as shown in Tables 1 and 2. For example, $\mathsf{u}_1^{t_2}(SM, FCE) = -6$ *and* $\mathsf{u}_2^{t_3}(SM, FCE) = -6$.

4. Main Solution Concepts in Simultaneous Move Games

In this section we will recap some existing solution concepts from static (simultaneous move) game with reference to our definition of an SDSRA security game. These solutions concepts include the Nash Equilibrium, the mixed Bayes-Nash Equilibrium (BNE), the maximin strategy, and the minimax regret strategy.

Nash Equilibrium: A Nash Equilibrium of a two player non-cooperative static (simultaneous move) game is a mixed strategy profile (Δ_1^*, Δ_2^*) such that each player is playing a best response to the strategy selected by their opponent. Formally, a Nash Equilibrium is a mixed strategy profile (Δ_1^*, Δ_2^*) which satisfies that, for each player $i \in N$ and for any mixed strategy Δ_i:

$$\mathsf{u}_1(\Delta_1^*, \Delta_2^*) \geq \mathsf{u}_1(\Delta_1, \Delta_2^*), \tag{2}$$

$$\mathsf{u}_2(\Delta_1^*, \Delta_2^*) \geq \mathsf{u}_2(\Delta_1^*, \Delta_2). \tag{3}$$

Example 3. *Consider the security game from Table 3. Suppose the column player is a defender and the row player is an attacker where the attacker selects an equilibrium*

strategy mixed strategy $\{\Delta_2(X) = q, \Delta_2(Y) = 1 - q\}$ *such that* $0 \leq q \leq 1$. *In this case, the defender's expected utility will be:*

$$\mathsf{u}_1(\Delta_1, \Delta_2) = (p \cdot q \cdot 1) + (p \cdot (1-q) \cdot 0) + ((1-p) \cdot q \cdot 0) + ((1-p) \cdot (1-q) \cdot 1)$$

for any mixed strategy $\{\Delta_1(A) = p, \Delta_1(B) = 1 - p\}$ *such that* $0 \leq p \leq 1$.

Thus, by formulas (2) and (3), the defender's best response to the strategy selected by the attacker is: if $q < \frac{1}{2}$, *then* $p = 0$; *if* $q > \frac{1}{2}$, *then* $p = 1$; *and if* $q = \frac{1}{2}$, *then* $p \in [0, 1]$. *Similarly, the attacker's best response to the strategy selected by the defender is: if* $p < \frac{3}{4}$, *then* $q = 1$; *if* $p > \frac{3}{4}$, *then* $q = 0$; *and if* $p = \frac{3}{4}$, *then* $q \in [0, 1]$. *Thus, the unique Nash Equilibrium* (Δ_1, Δ_2) *of this game that satisfies formulas (2) and (3) is* $\{\Delta_1(A) = \frac{3}{4}, \Delta_1(B) = \frac{1}{4}\}$ *and* $\{\Delta_2(X) = \frac{1}{2}, \Delta_2(Y) = \frac{1}{2}\}$.

Mixed Bayes-Nash Equilibrium (BNE): The mixed BNE is a solution concept for a non-cooperative static game with incomplete information about the characteristics (*i.e.*, utility values) of other players. In this solution concept, each player is assumed to know the equilibrium mixed strategies of other players as well as the possible player types of other players. In addition, none of the players can obtain a higher expected utility by only changing their own mixed strategy. More formally, in an SDSRA security game $(N, T, \mathsf{p}, K, A, \Psi, U)$, the mixed BNE with respect to each attacker type t is a mixed strategy profile $(\Delta_1^*, \Delta_2^{t,*})$ which satisfies that, for any mixed strategy Δ_1 of the defender, we have that:

$$\sum_{t \in T} \mathsf{p}(t) \mathsf{u}_1^t(\Delta_1^*, \Delta_2^{t,*}) \geq \sum_{t \in T} \mathsf{p}(t) \mathsf{u}_1^t(\Delta_1, \Delta_2^{t,*}). \tag{4}$$

Also, for any mixed strategy Δ_2^t of the attacker of type t, we have that:

$$\mathsf{u}_2^t(\Delta_1^*, \Delta_2^{t,*}) \geq \mathsf{u}_2^t(\Delta_1^*, \Delta_2^t). \tag{5}$$

Clearly, by formulas (4) and (5), the computation of a mixed BNE is similar to the Nash Equilibrium in Example 3. In additional, in order to satisfy the assumption that each player knows the equilibrium mixed strategies of other players, it is assumed that the utility matrices are common knowledge[2] to all players. Particularly, if there exists a mixed BNE for a given static game in which each player needs to assign a positive probability for each pure strategy, then we call this mixed BNE a *completely* mixed BNE.

Maximin strategy: The concept of loss-aversion is based on the observation that decision makers will strongly prefer to avoid unacceptable losses when faced with uncertainty,

[2]There is common knowledge of ϕ in a group of agents when all agents know ϕ, when all agents know that all agents know ϕ, and so on ad infinitum. (Fagin et al., 2004)

rather than to attempt to acquire higher gains. In decision theory, this concept is usually formalized by the maximin decision rule (*i.e.*, Γ-maximin) (Osborne, 2003). In game theory, the maximin strategy is a solution strategy based on this concept, in which a player seeks to minimize potential losses by considering the worst case (maximum loss) scenario. More formally, with the defender (*i.e.*, player 1) and the attacker (*i.e.*, player 2) in an SDSRA security game, the maxmin strategy $\underline{\Delta_i}$ for player i ($i \in \{1, 2\}$) is defined as follows:

$$\underline{\Delta_1} = \underset{\Delta_1}{\operatorname{argmax}} \left(\min_{\Delta_2} \mathsf{u}_1(\Delta_1, \Delta_2) \right), \tag{6}$$

$$\underline{\Delta_2} = \underset{\Delta_2}{\operatorname{argmax}} \left(\min_{\Delta_1} \mathsf{u}_1(\Delta_1, \Delta_2) \right). \tag{7}$$

Example 4. *Consider the security game from Table 3. Suppose the defender's maximin strategy is* $\{\underline{\Delta_1}(A) = p, \underline{\Delta_1}(B) = 1 - p\}$ *with* $0 \leq p \leq 1$. *Since there does not exist any dominant strategy in this game, by formulas (6) and (7), there exists a constant* C *such that for any mixed strategy* $\{\Delta_2(X) = q, \Delta_2(Y) = 1 - q\}$ *with* $0 \leq q \leq 1$, *we have*

$$\begin{aligned} \mathsf{u}_1(\underline{\Delta_1}, \Delta_2) &= (p \cdot q \cdot 1) + (p \cdot (1-q) \cdot 0) + ((1-p) \cdot q \cdot 0) + ((1-p) \cdot (1-q) \cdot 1) \\ &= C. \end{aligned}$$

Thus, for the defender, we have $\{\underline{\Delta_1}(A) = \frac{1}{2}, \underline{\Delta_1}(B) = \frac{1}{2}\}$. *Similarly, for the attacker, by the same method, we have* $\{\underline{\Delta_2}(X) = \frac{1}{4}, \underline{\Delta_2}(Y) = \frac{3}{4}\}$.

Minimax regret strategy: Regret is an emotion associated with decisions which yield less desirable outcomes. In decision theory, this concept is usually formalized by the principle of minimax regret (Savage, 1951). As with loss tolerance, this concept can be applied into game theory. That is, when a player selects a strategy, they may feel regret when considering the possibility of obtaining higher utilities by selecting alternative strategies. In order to limit regret, the player should select a strategy that minimizes their maximum regret with respect to all other strategies. More formally, with the defender (*i.e.*, player 1) and the attacker (*i.e.*, player 2) in an SDSRA security game, the minimax regret strategy $\gamma(\Delta_i)$ for player $i \in N$ is defined as follows:

$$\gamma(\Delta_1) = \underset{\Delta_1}{\operatorname{argmin}} \left(\max_{\Delta_2} \left(\max_{\Delta_1'} \mathsf{u}_1(\Delta_1', \Delta_2) - \mathsf{u}_1(\Delta_1, \Delta_2) \right) \right), \tag{8}$$

$$\gamma(\Delta_2) = \underset{\Delta_2}{\operatorname{argmin}} \left(\max_{\Delta_1} \left(\max_{\Delta_2'} \mathsf{u}_2(\Delta_1, \Delta_2') - \mathsf{u}_2(\Delta_1, \Delta_2) \right) \right). \tag{9}$$

Example 5. *Consider the security game from Table 3. Suppose the attacker's minimax regret strategy is* $\{\gamma(\Delta_2)(X) = q, \gamma(\Delta_2)(Y) = 1 - q\}$ *with* $0 \leq q \leq 1$. *Since there does not exist any dominant strategy in this game, by formulas (8) and (9), for any pure strategy (*i.e.*,* A *and* B*) of the defender, we have*

$$\begin{aligned} u_2(A, Y) - u_2(A, \gamma(\Delta_2)) &= 1 - 1 \times (1 - q) \\ &= u_2(B, X) - u_2(B, \gamma(\Delta_2)) \\ &= 3 - 3 \times q. \end{aligned}$$

Thus, for the attacker, we have $\{\underline{\Delta_2}(X) = \frac{3}{4}, \underline{\Delta_2}(Y) = \frac{1}{4}\}$. *Similarly, for the defender, by the same method, we have* $\{\underline{\Delta_1}(A) = \frac{1}{2}, \underline{\Delta_1}(B) = \frac{1}{2}\}$.

In this section, we have provided some simple examples for the computation of all related solution concepts, including the Nash Equilibrium, the Mixed BNE, the Maximin strategy, and the Minimax regret strategy. For more details, as well as some approximate algorithms, we refer the reader to (Osborne, 2003).

5. Rationalizability in SDSRA

In this section, we will provide an epistemic characterization of the intuition behind our solution concept (*i.e.*, acceptable costs of minimax regret equilibrium), comparable to the idea of rationalizability in the Nash Equilibrium (Osborne, 2003). Recall the assumptions from Section 1 on which our solution concept is based:

(A1) Each player considers the influence of loss-aversion (*i.e.*, the tendency to prefer ensuring a sufficient minimum expected utility, rather than seeking some potential maximum expected utility).

(A2) Each player minimizes their maximum regret while considering their attitude towards loss-aversion and the strategic choices of others.

(A3) The attacker's utility matrix is known by the defender and each player knows their own utility matrix.

Clearly, these assumptions differ from Nash Equilibrium or mixed BNE, which generally assume that each player always seeks to maximize their expected utility based on common knowledge (*e.g.*, the utilities and expected behavior of other players). The reason why we remove the traditional assumptions are: (i) common knowledge about the beliefs of others can lead to an infinite hierarchy of beliefs which may in turn be intractable for real-world applications, particularly in the case of online SDSRA systems; (ii) it is

not always realistic to assume that an attacker has the correct subjective beliefs about a defender's strategy, because they cannot know the defender's thought about the attacker type; and (iii) the defender's prediction of the attacker's strategy may be imperfect but, since some losses are unacceptable, they want to ensure that, at least, their losses will not exceed an acceptable minimal payoff. In term of "acceptable" means that, depending on the player's attitude towards loss-aversion, the player only selects a strategy with a loss (*i.e.*, a negative payoff) that is higher than some given constant. Therefore, this constant can be viewed as the minimal *acceptable* payoff, *i.e.*, the constant acts as a threshold to determine whether or not a strategy is acceptable to the player.

Consider assumption **A1**: *Each player considers the influence of loss-aversion.* The meaning of this assumption is that, if a player cannot determine a strategy with maximum expected utility with respect to the other player's strategy, then they should strongly prefer to avoid unacceptable losses rather than attempt to acquire higher gains. The influence of loss-aversion on decision making has been convincingly demonstrated in psychological and economic experiments (Kahneman, 2003). Also, the loss-aversion coefficients may vary for different categories of decision maker (Kuhnen and Knutson, 2005). Some decision makers are willing to suffer a higher loss for the chance of acquiring higher rewards, whilst some decision makers strongly prefer avoiding losses to acquiring gains. In SD-SRA problems, both players suffer from the same problem. If a player chooses a strategy that causes the *maximin* expected utility, then they can ensure that their expected utility is no less than this value. Therefore, the selection of another strategy always means that their expected utility is less than the maximin expected utility. That is, players suffer a loss of their minimum expected utility if they select a different strategy. This idea of considering a player's attitude towards loss-aversion has been discussed in the literature. An example in (Pruzhansky, 2011) called Aumann and Maschler Game is, perhaps, the most well-known. Consider the Aumann and Maschler Game in Table 3. By Example 3, we can find that the unique Nash Equilibrium of this game is $\{\Delta_{\text{row}}(A) = \frac{3}{4}, \Delta_{\text{row}}(B) = \frac{1}{4}\}$ and $\{\Delta_{\text{col}}(X) = \frac{1}{2}, \Delta_{\text{col}}(Y) = \frac{1}{2}\}$. As such, by formula (1), the expected utilities are $\frac{1}{2}$ for the row player and $\frac{3}{4}$ for the column player. Moreover, by Example 4, we can find that the maximin strategy for the row player is $\{\Delta_{\text{row}}(A) = \frac{1}{2}, \Delta_{\text{row}}(B) = \frac{1}{2}\}$ and the maximin strategy for the column player is $\{\Delta_{\text{col}}(X) = \frac{1}{4}, \Delta_{\text{col}}(Y) = \frac{3}{4}\}$. In this case, by formula (1), the worst case expected utility for the row player is $\frac{1}{2}$ and for the column player is $\frac{3}{4}$. As a result, the expected utilities for each player with respect to the Nash Equilibrium can be guaranteed by the maximin strategy for each player.

In the work of Harsanyi (1977) and Pruzhansky (2011), there are many arguments about what strategy should be selected by each player: Nash Equilibrium or maximin strategies? Some researchers, such as Harsanyi (Harsanyi, 1977), have argued that the players should choose their maximin strategies, since the Nash Equilibrium means a player

	X	Y
A	40, 320	80, 40
B	80, 40	40, 80

Table 4: Asymmetric matching pennies.

risks losing their maximin value without gaining a higher expected utility. Without losing the general idea of loss-aversion, in our games, we can interpret the meaning of loss-aversion more specifically as follows:

> *A player selects a strategy with a higher gain, as long as the minimum expected utility for this strategy is an acceptable reduction of their maximin expected utility.*

In line with the notion of an acceptable strategy, the term of *acceptable reduction* refers a requirement that the difference between the minimum expected utility and the maximin expected utility is less than the difference between the acceptable minimal payoff and the maximin expected utility.

While the maximin strategy can guarantee the attacker's minimum expected utility, it is possible that it cannot be represent a correct prediction of the behavior of the attacker in an SDSRA security game, (*e.g.*, the attacker may choose to risk an attempt to obtain a higher expected utility). An experimental evaluation of the game in Table 4 was carried out in (Goeree et al., 2001). In this game, by formulas (4) and (5), following the same method in Example 3, we can obtain that the unique Nash Equilibrium is $(\Delta_{\text{row}}, \Delta_{\text{col}})$ such that $\{\Delta_{\text{row}}(A) = \frac{1}{8}, \Delta_{\text{row}}(B) = \frac{7}{8}\}$ and $\{\Delta_{\text{col}}(X) = \frac{1}{2}, \Delta_{\text{col}}(Y) = \frac{1}{2}\}$. On the other hand, by formulas (6) and (7), following the same method in Example 4, the maximin strategy for the column player is the mixed strategy $\{\Delta_{\text{col}}(X) = \frac{1}{8}, \Delta_{\text{col}}(Y) = \frac{7}{8}\}$. By formula (1), we can easily find that the maximin strategy for the column player can guarantee that their expected utility is not less than the expected utility for the completely mixed Nash Equilibrium. Thus, if the column player must select a pure strategy to play, then he should select X and Y in an even number of cases with the Nash Equilibrium, but Y in most cases with the maximin strategy. However, the experimental results in (Goeree et al., 2001) show quite different results for the column player if most people (96%) were to choose a pure strategy X. This is our reason for considering other factors that may influence the attacker's decision in an SDSRA security game.

Consider assumption **A2**: *Each player minimizes their maximum regret while considering their attitude towards loss-aversion and the strategic choices of others.* The meaning of this assumption is that, in our games, players minimize their maximum regret based on an acceptable minimal payoff, rather than maximize their expected utility based on their

	X	Y
A	40, 350	80, 80
B	80, -190	40, 80

Table 5: Regret game.

subjective beliefs about another player's strategy. Actually, since the minimax regret strategy does not require a player to have knowledge of the other players' strategies, to some extent, it is more realistic to select this strategy than the Nash Equilibrium strategy in an online surveillance environment. In fact, many behavioral studies (*e.g.*, (Chua et al., 2009)) show that human decisions under uncertainty are strongly influenced by the emotion of regret. The *minimax regret* principle suggested in (Savage, 1951) says that a choice is admissible if this choice minimizes the maximum difference between the outcome of a choice and the best outcome that could potentially have been obtained. For the game in Table 4, by formulas (8) and (9), following the same method in Example 5, we find that the minimax regret strategy for the column player is $\{\Delta_{\text{col}}(X) = \frac{7}{8}, \Delta_{\text{col}}(Y) = \frac{1}{8}\}$. This result supports the findings from (Goeree et al., 2001), which says that most people will select the pure strategy X. On the other hand, in the field of game theory, researchers increasingly consider the effect of regret in strategy selections. For example, Halpern and Pass (2012) applied the minimax regret principle to explain many paradoxes in game theory that are caused by applying the Nash Equilibrium concept, such as the well-known Traveler's Dilemma and the Centipede Game. However, in an SDSRA problem, different defenders might have different degrees of loss-aversion. Since an attacker cannot fully observe a defender's whole strategy selection process, it is unrealistic to assume that they have perfect knowledge of a defender's attitude towards loss-aversion.[3] For this reason, the Iterated Regret Minimization method in (Halpern and Pass, 2012) cannot be applied into our surveillance environment.

Although there are many advantages of the minimax regret strategy (*e.g.*, in producing satisfactory results for many experimental behavioral studies), it still has some limitations in addressing the problem of online SDSRA. Let us discuss the game in Table 5. By formulas (8) and (9), following the same method in Example 5, we can find that the minimax regret strategy for the column player is $\{\Delta_{\text{col}}(X) = \frac{1}{2}, \Delta_{\text{col}}(Y) = \frac{1}{2}\}$. However, selecting this mixed strategy means that the column player may suffer a negative expected utility -55 if the row player selects the strategy B. Conversely, the column player can guarantee

[3]Conversely, an attacker's degree of loss-aversion can be obtained from historical data, criminology experts, etc.

the positive expected utility of 80 by selecting pure strategy Y. In the security domain, clearly it is more reasonable that a strategy which reduces the potential loss is preferred to a strategy which may result in higher gain, *e.g.*, if the loss is unacceptable, such as human life. As such, it is also necessary to consider a player's degree of loss-aversion. In this vein, we rely on the maximin strategy for determining an acceptable minimal payoff, and suggest that players should seek to minimize their maximum regret based on this acceptable minimal payoff.

Finally, consider assumption **A3**: *The attacker's utility matrix is known by the defender and each player knows their own utility matrix.* This is already accepted by solution concepts in the Stackelberg game framework (Tambe, 2011) because, when being applied to real-world security applications, this assumption is more realistic than that of the Nash Equilibrium. There are two obvious reasons. The first reason is that **A3** allows the attacker to have partial information about the defender's utility matrix, whereas in the Nash Equilibrium this information must be prefect. In real-world application, although an attacker may be aware of the defender's utility values associated with attacks on some targets, it is not very realistic to assume that an attacker has perfect information about all targets. For example, some utility values may correspond to subjective estimations by a domain expert, and others has may correspond to dynamic features of the environment. For instance, the utility value of a unique location, (*e.g.*, the Butterfly Garden in Singapore Changi Airport) might be subjectively estimated by the domain expert, while the utility value of a shopping mall might be determined by the number of people inside (which will fluctuate dynamically). On the other hand, utility values may be updated at any time due to external events that the attacker may not realise. The second reason is that **A3** does not require that strategies and utility values are common knowledge among players, whereas this is required by the Nash Equilibrium. As mentioned previously, it is unlikely that players in an online SDSRA environment can be sure about the information obtained by their opponent, thus common knowledge about the beliefs of others is impossible.

These assumptions reveal two factors which must be considered during a player's strategy selection: (i) guaranteed safety level—a given player compares the minimum expected utility of their selected strategy with the maximin expected utility in the game, especially in situations where this player does not have common knowledge of the opponent's strategy or where the expected utility for each strategy is imprecise; and (ii) maximum regret minimization—a given player attempts to minimize the anticipated emotion of regret under uncertainty by a (hypothetical) comparison between the minimum expected utility of a strategy and the maximum expected utility that alternatives may provide. Thus, according to assumptions **A1**, **A2**, and **A3**, our solution concept should follow the principle below:

> *A player will select a strategy with a lower maximum regret, after considering whether or not its minimum expected utility is an acceptable reduction of their*

maximin expected utility.

By considering the acceptable reduction of their maximin expected utility as an acceptable cost, we call it the principle of acceptable costs of minimax regret. This principle has two advantages. *Firstly*, it avoids the overly pessimistic result of the maximin strategy, which focuses on the worst case outcome. For example, suppose a lottery game sells \$1 per-ticket with a 99% chance of winning \$5,000, then the maximin strategy would reject the offer, since the potential loss of \$1 could lead to the minimum value lower than rejecting the offer. However, following our principle, the regret for not buying the ticket is \$4,999 and the regret for buying the ticket is \$1 while the maximin expected utility is not strictly required. As such, if losing \$1 is acceptable to a player, then they will take the risk. *Secondly*, it avoids the potential for unacceptable losses with the minimax regret strategy. For example, suppose a lottery sells \$100 per-ticket with a 1% chance of winning \$5,000, then the minimax regret strategy would always accept the offer. However, in our principle, since a player needs to consider whether losing \$100 is an acceptable cost, they may not accept the offer. These advantages are useful in real-world applications since, in security domains, some losses are unacceptable (*e.g.*, human life) while a defender may lose the chance to act by selecting an overly pessimistic strategy. Based on this principle, we call our solution concept the *acceptable costs of minimax regret equilibrium, which will be detailed in the next section.*

6. Acceptable Costs of Minimax Regret Equilibrium

In order to determine each player's strategy according to the principle of acceptable costs of minimax regret, we must first determine an acceptable minimal payoff. Clearly, the maximum value of a player's acceptable minimal payoff is their maximin expected utility (*i.e.*, $\max\min(X)$), while the minimum value is their minimin expected utility (*i.e.*, $\min\min(X)$). Thus, their acceptable minimal payoffs must be in the interval $[\min\min(X), \max\min(X)]$. In order to support different degrees of loss-aversion for different types of players, we rely on a similar approach to that used by the Hurwicz criterion (Jaffray and Jeleva, 2007) by transforming this interval into a point value using a loss tolerance degree $\sigma \in [0, 1]$. With this degree, we can determine the acceptable minimal payoff for each player by $\sigma \min\min(X) + (1 - \sigma) \max\min(X)$.

In other words, the player needs to consider the lower and upper boundaries of their minimum expected utility and, according to their loss tolerance degree, will place more or less importance on each. Clearly, a higher σ value will result in a lower acceptable minimal payoff and a more risk-seeking player. In particular, if $\sigma = 1$, then a player does not care about potential loss (*i.e.*, the player is completely optimistic). Conversely, if $\sigma = 0$, then a player cares only about potential loss (*i.e.*, the player will be completely pessimistic). In

real-world applications, the issue of estimating loss tolerance degrees is arguably no more or less realistic than estimating utility values or the probability distribution over attacker types. Essentially, we are assuming that this information can be learned from data or can be estimated by domain experts. In fact, even if it is impossible to estimate these values, our method is still capable of modelling the concept of loss tolerance in a general way, through a fixed loss tolerance degree for all players. For example, if we want to state that all players are neutral about the loss then we might assign a loss tolerance degree of 0.5 to all players.

Once we have identified the strategies which are acceptable according to a player's loss tolerance degree, the remaining issue is to ensure that the player selects, from those strategies, the strategy with the lowest maximum regret. Now we will discuss how to predict the strategy that will be selected by each type of attacker according to our principle, before describing how to select the defender's optimal strategy according to this prediction.

Firstly, we formally define the concept of maximum regret as follows:

Definition 2. *The maximum regret for an attacker with type t over a strategy Δ_2, denoted $r^t(\Delta_2)$, is defined as:*

$$r^t(\Delta_2) = \max_{a_1} \left(\max_{\Delta_2'} \mathsf{u}_2^t(a_1, \Delta_2') - \mathsf{u}_2^t(a_1, \Delta_2) \right). \tag{10}$$

In Definition 2, by formula (1), given the linearity of utility functions and the fact that there does not exist any dominant strategy for the defender, we can directly find that for any mixed strategy of the attacker, there always exists a pure strategy a_1 of the defender such that

$$\left(\max_{\Delta_2'} \mathsf{u}_2^t(a_1, \Delta_2') - \mathsf{u}_2^t(a_1, \Delta_2) \right) \geq \left(\max_{\Delta_2'} \mathsf{u}_2^t(\Delta_1, \Delta_2') - \mathsf{u}_2^t(\Delta_1, \Delta_2) \right).$$

This is the reason why we only consider the pure strategies of the defender in Definition 2. Hence, this definition allows us to predict the optimal strategy for each type of attacker as follows:

Definition 3. *Let $\sigma_2^t \in [0, 1]$ be the loss tolerance degree for attacker type t and Ω_2 be a set of mixed strategies of the attacker. Then the optimal mixed strategy for t, denoted $\Delta_2^{t,*}$, is defined as:*

$$\Delta_2^{t,*} = \operatorname*{argmin}_{\Delta_2 \in \Omega_2} r^t(\Delta_2), \tag{11}$$

such that for any mixed strategy $\Delta_2 \in \Omega_2$, we have

$$\min_{a_1} \mathsf{u}_2^t(a_1, \Delta_2) \geq \left(\max_{\Delta_2'} \min_{a_1'} \mathsf{u}_2^t(a_1', \Delta_2') \right) - \varsigma_2^t, \tag{12}$$

$$\varsigma_2^t = \sigma_2^t \left(\max_{\Delta_2'} \min_{a_1'} \mathsf{u}_2^t(a_1', \Delta_2') - \min_{\Delta_2''} \min_{a_1''} \mathsf{u}_2^t(a_1'', \Delta_2'') \right). \tag{13}$$

Formula (10) means that an attacker will select, as their optimal strategy, a mixed strategy which can minimize their maximum regret. However, formula (12) imposes a constraint on the possible mixed strategies, which can be selected depending on whether or not the strategy has an acceptable cost. That is, the minimum expected utility of the strategy should be higher than the acceptable minimal payoff. formula (13) shows how to calculate the maximum acceptable reduction, where ς_2^t denotes the maximum loss that attacker type t can tolerate given their loss tolerance degree σ_2^t. Clearly, some attacker types may accept a choice with a lower minimum expected utility to reduce their maximum regret, while some may reject a higher loss of their minimum expected utility. For example, a terrorist usually shows higher tolerance for loss of their minimum expected utility than an armed robber, who is commonly more risk-averse. In real-world applications, the value for σ_2^t can be obtained for each type of attacker from, for example, historical data, criminology experts, and so on. Finally, by formulas (12) and (13), the acceptable minimal payoff of the attacker with a given attacker type can be presented as follows:

$$\begin{aligned} & \max_{\Delta_2'} \min_{a_1'} \mathsf{u}_2^t(a_1', \Delta_2') - \sigma_2^t \left(\max_{\Delta_2'} \min_{a_1'} \mathsf{u}_2^t(a_1', \Delta_2') - \min_{\Delta_2''} \min_{a_1''} \mathsf{u}_2^t(a_1'', \Delta_2'') \right) \\ = \; & \sigma_2^t \min_{\Delta_2''} \min_{a_1''} \mathsf{u}_2^t(a_1'', \Delta_2'') + (1 - \sigma_2^t) \max_{\Delta_2'} \min_{a_1'} \mathsf{u}_2^t(a_1', \Delta_2'). \end{aligned}$$

Now we can discuss how to find an optimal strategy for the defender based on the predicted mixed strategy $\Delta_2^{t,*}$ for each type of attacker and the probability distribution over the set of attacker types. Following the principle of acceptable costs of minimax regret, we can define the defender's optimal strategy as follows:

Definition 4. *Let $\sigma_1 \in [0, 1]$ be the loss tolerance degree for the defender and $\Delta_2^{t,*}$ be the optimal mixed strategy for attacker type t. Then the optimal strategy for the defender, denoted Δ_1^*, is defined as:*

$$\Delta_1^* = \operatorname*{argmax}_{\Delta_1} \sum_{t \in T} \mathsf{p}(t) \mathsf{u}_1^t(\Delta_1, \Delta_2^{t,*}), \tag{14}$$

such that for any pure strategy a_1 that has a positive probability value in Δ_1, we have

$$\min_{a_2} \sum_{t \in T} \mathsf{p}(t) \mathsf{u}_1^t(a_1, a_2) \geq \left(\max_{a_1'} \min_{a_2} \sum_{t \in T} \mathsf{p}(t) \mathsf{u}_1^t(a_1', a_2) \right) - \varsigma_1, \tag{15}$$

$$\varsigma_1 = \sigma_1 \left(\max_{a_1'} \min_{a_2} \sum_{t \in T} \mathsf{p}(t) \mathsf{u}_1^t(a_1', a_2) - \min_{a_1''} \min_{a_2} \sum_{t \in T} \mathsf{p}(t) \mathsf{u}_1^t(a_1'', a_2) \right). \quad (16)$$

The reason why we adopt the maximum expected utility in formula (14) is that the defender already knows the attacker's optimal mixed strategy $\Delta_2^{t,*}$ and the probability distribution over the attacker's possible types. So, according to Assumption A2 and by Definition 3, the minimax regret strategy is the same as the maximum expected utility strategy for the defender. Moreover, since the attacker's strategy is based on a judgement of the attacker's utility matrix, the loss tolerance assumption for each type of attacker, and imperfect information obtained by surveillance system, there is a chance that the attacker may play a different strategy than the strategy predicted by the defender. Thus, by the linearity of the utility function, we only need to consider the pure strategies that may be selected by the attacker in formulas (15) and 16. Hence, with regards to the σ_1 value, a security manager can fine-tune this value to reflect different real-time applications. In this way, our method is more flexible in balancing the possibility of unacceptable losses caused by the failure of prevention and the expected utility for successfully preventing an attack. Finally, by formulas (15) and (16), the acceptable minimal payoff of the attacker with a given attacker type can be presented as follows:

$$\begin{aligned} & \max_{a_1'} \min_{a_2} \sum_{t \in T} \mathsf{p}(t) \mathsf{u}_1^t(a_1', a_2) - \sigma 1 \left(\max_{a_1'} \min_{a_2} \sum_{t \in T} \mathsf{p}(t) \mathsf{u}_1^t(a_1', a_2) - \min_{a_1''} \min_{a_2} \sum_{t \in T} \mathsf{p}(t) \mathsf{u}_1^t(a_1'', a_2) \right) \\ = \; & (1 - \sigma_1) \max_{a_1'} \min_{a_2} \sum_{t \in T} \mathsf{p}(t) \mathsf{u}_1^t(a_1', a_2) + \sigma_1 \min_{a_1''} \min_{a_2} \sum_{t \in T} \mathsf{p}(t) \mathsf{u}_1^t(a_1'', a_2). \end{aligned}$$

6.1. Properties

Since the correctness of the defender's optimal strategy in our method is based on a prediction of the attacker's strategy, we consider the properties of Definition 3 to justify the our prediction of the attacker's strategy. Moreover, we will show that the defender's strategy in our solution concept can be considered as an extension to the maximum expected utility strategy and that considering the defender's set of pure strategies is sufficient to find the solution. Given these properties, the whole process in our solution concept (*i.e.*, acceptable costs of minimax regret equilibrium) can be interpreted as an optimization problem for which there exists efficient methods of computation.

Theorem 1. *Let $A_1 = \{a_1^1, \ldots, a_1^n\}$ be the set of pure strategies for the defender, $A_2 = \{a_2^1, \ldots, a_2^n\}$ be the set of pure strategies for the attacker and Δ_2 be a mixed strategy for the attacker such that $\Delta_2(a_2^i) = q_i$ for each $i \in 1, \ldots, n$. Then the attacker's maximin strategy in an SDSRA security game is a unique equalizer. Here, for a two player game,*

an equalizer is a mixed strategy Δ_i of a player $i \in N$ such that there exists a constant $c \in \mathbb{R}$ where for any pure strategy $a_j \in A_j$ of another player j ($j \neq i$), we have that $u_i(\Delta_i, a_j) = c$.

Proof. By the condition of an equalizer, our game has a unique equalizer if and only if for linear equation $Aq = u$, where

$$A = \begin{bmatrix} \mathsf{u}_2(a_1^1, a_2^1) & \cdots & \mathsf{u}_2(a_1^1, a_2^n) \\ \vdots & \ddots & \vdots \\ \mathsf{u}_2(a_1^n, a_2^1) & \cdots & \mathsf{u}_2(a_1^n, a_2^n) \\ 1 & \cdots & 1 \end{bmatrix}, q = \begin{bmatrix} q_1 \\ \vdots \\ q_n \end{bmatrix}, u = \begin{bmatrix} c \\ \vdots \\ c \\ 1 \end{bmatrix},$$

there exists a unique solution q. Thus, $\mathsf{rank}(A) = n$.[4] In other words, it requires: (i) that no convex combination of some rows in A dominate convex combinations of other rows; and (ii) that the utility matrix satisfies $|A_1| = |A_2| = n$. As mentioned after Definition 1, item (i) holds because there does not exist any dominated strategy for the attacker in an SDSRA security game. Similarly, item (ii) we holds by the definition of an SDSRA security game. □

This theorem reveals that an attacker can always find a unique strategy that guarantees their expected utility regardless of any mixed strategy of the defender.

Theorem 2. *In an SDSRA security game, for each type of attacker, the expected utility of the maximin strategy will not be less than a completely mixed BNE in the game.*

Proof. Suppose the game is a simultaneous 2-player game, in which each player has a finite set of strategies, (Δ_1^*, Δ_2^*) is a completely mixed equilibrium with excepted utilities u_1^* and u_2^* for players 1 and 2 respectively, and $(\underline{\Delta_1}, \underline{\Delta_2})$ is the maximin strategy with minimum expected utilities $\underline{u_1}$ and $\underline{u_2}$ respectively. By Theorem 1, we can obtain $\underline{\Delta_2}$ is an equalizer directly. Moreover, by the proof in (Pruzhansky, 2011) and the fact that $\underline{\Delta_2}$ is an equalizer, we have that $\underline{u_2} = u_2^*$. □

Since our games satisfy that no pure or mixed strategy of an attacker is strictly or weakly dominated by a convex combination of their other strategies, Theorem 2 shows that the attacker can always guarantee that their expected utility is no less than the completely mixed BNE by selecting a maximin strategy. Thus, these theorems demonstrate that it is reasonable for each type of attacker to quantify losses in relation to their maximin strategy.

[4] In linear algebra, $\mathsf{rank}(A)$ denotes the rank of a matrix A, *i.e.*, the dimension of the column space of A.

Theorem 3. *The maximum regret $r^t(\Delta_2)$ for the attacker of type t for the strategy Δ_2 in an SDSRA security game can be obtained as follows:*

$$r^t(\Delta_2) = \max_{a_1} \left(\max_{a_2} \mathsf{u}_2^t(a_1, a_2) - \mathsf{u}_2^t(a_1, \Delta_2) \right).$$

Proof. Given the linearity of utility functions and the fact that there does not exist any dominated strategy for the attacker, we have this result straightforward. □

This theorem means that we only need to consider the attacker's pure strategies when determining their maximum expected utility strategy with respect to the defender's strategy. In other words, this theorem confirms that our method improves the efficiency of predicting the strategy of each type of attacker.

Theorem 4. *Suppose the utility value for successfully attacking any target is the same for an attacker of type t,* i.e.*, for any pure strategy profile $(a_1, a_2) \in A_1 \times A_2$ that satisfies $a_1 \neq a_2$ (a_1 is a pure strategy selected by the defender and a_2 is a pure strategy selected by the attacker), there exist a constant $c \in \Re^+$, such that $\mathsf{u}_2^t(a_1, a_2) = c$.[5] Then the minimax regret strategy is the same as the maximin strategy for the attacker.*

Proof. Suppose the mixed strategy $\Delta_2^{t,\uparrow}$ is the maximin strategy for attacker type t. By Theorem 1, for any defender's pure strategy a_1, there exists a constant k such that $\mathsf{u}_2^t(a_1, \Delta_2^{t,\uparrow}) = k$. Then, by Theorem 3, Definition 2, $\mathsf{u}_2^t(a_1, a_2) = c$ if $a_1 \neq a_2$, and the fact that attacker only wins in the case where the defender selects a different target from the attacker, we have:

$$r^t(\Delta_2^{t,\uparrow}) = c - k. \tag{17}$$

Suppose there exists a minimax regret strategy $\Delta_2^{t,*} \neq \Delta_2^{t,\uparrow}$, then by Definition 2, we have $r^t(\Delta_2^{t,*}) \leq r^t(\Delta_2^{t,\uparrow})$.

First, we consider the case that $r^t(\Delta_2^{t,*}) = r^t(\Delta_2^{t,\uparrow})$. By Theorem 1, Definition 2, and the fact that $\mathsf{u}_2^t(a_1, a_2) = c$ for any $a_1 \neq a_2$, we have $\Delta_2^{t,*} = \Delta_2^{t,\uparrow}$. This violates our assumption that $\Delta_2^{t,*} \neq \Delta_2^{t,\uparrow}$. Thus, we have $r^t(\Delta_2^{t,*}) \neq r^t(\Delta_2^{t,\uparrow})$.

Second, we consider the case that $r^t(\Delta_2^{t,*}) < r^t(\Delta_2^{t,\uparrow})$. Since $\Delta_2^{t,\uparrow}$ is a maximin strategy, there always exists a defender's pure strategy a_1', such that $\mathsf{u}_2^t(a_1', \Delta_2^{t,*}) < \mathsf{u}_2^t(a_1', \Delta_2^{t,\uparrow})$. Then, by Theorem 1, Definition 2 and formula (17), we have

$$r^t(\Delta_2^{t,\uparrow}) = c - k = c - \mathsf{u}_2^t(a_1', \Delta_2^{t,\uparrow}).$$

Thus, for the defender's pure strategy a_1', we have $c - \mathsf{u}_2^t(a_1', \Delta_2^{t,*}) > r^t(\Delta_2^{t,\uparrow})$. This violates our assumption that $r^t(\Delta_2^{t,*}) < r^t(\Delta_2^{t,\uparrow})$.

Therefore, according to the conclusions of these two cases, we have $\Delta_2^{t,*} = \Delta_2^{t,\uparrow}$. □

[5] Here $a_1 \neq a_2$ means that the defender and the attacker select different targets and thus the attacker wins.

Theorem 4 demonstrates that if the expected utility for successfully attacking each target is the same as that for a given attacker type, then they can choose their maximin strategy to guarantee their minimum expected utility as well as to reduce their maximum regret. This result is useful when considering the behavior of attacker types that only care about the successful attacker. For example, a pickpocket may view all targets as equally desirable, in which case, they are likely to base their decision on which target provides the greatest chance of escape. Thus, the pickpocket ia completely lose-averse when the potential gain is the same for all targets. The relationship between Definition 3 and both the minimax regret strategy (Savage, 1951) (*i.e.*, formulas (8) and (9) and maximin strategy (Osborne, 2003) (*i.e.*, formulas 6 and 7), is as follows:

Theorem 5. *Let $\sigma_2^t \in [0,1]$ be the loss tolerance degree for an attacker of type t and $\Delta_2^{t,*}$ be the optimal mixed strategy for t according to the principle of acceptable costs of minimax regret:*

(i) If $\sigma_2^t = 1$, then $\Delta_2^{t,*}$ is an optimal choice according to the maximum regret strategy.

(ii) If $\sigma_2^t = 0$, then $\Delta_2^{t,*}$ is an optimal choice according to the maximin strategy.

Proof. (i) By formulas (12) and (13) and the fact that $\sigma_2^t = 1$, then a mixed strategy Δ_2 can be any mixed strategy for the attacker. Then, by formula (10) $\Delta_2^{t,*}$ is also an optimal choice according to the minimax regret strategy. (ii) By formulas (10), (12), and (13) and the fact that $\sigma_2^t = 0$, then $\Delta_2^{t,*}$ can only be a mixed strategy with the maximin expected utility. Thus, $\Delta_2^{t,*}$ is also an optimal choice by the maximin strategy. □

Now we can consider a property of the defender's optimal strategy. Actually, Definition 4 can be seen as an extension to maximum expected utility theory by the following theorem:

Theorem 6. *If $\sigma_1 = 1$, then the defender's optimal pure strategy a_1^* is the maximum expected utility strategy.*

Proof. Suppose a_1^* is the defender's optimal strategy according to Definition 4. By formulas (15) and (16), and the fact that $\sigma_1 = 1$, we have $\forall a_1 \in A_1$, $\forall a_2 \in A_2$:

$$\sum_{t \in T} \mathsf{p}(t)\mathsf{u}_1^t(a_1, a_2) \geq \min_{a_1'} \min_{a_2} \sum_{t \in T} \mathsf{p}(t)\mathsf{u}_1^t(a_1', a_2).$$

As a result, we only need to find the pure strategy a_1 that satisfies formula (14). In other words, a_1^* is the strategy in the defender's set of pure strategies with maximum expected utility. Moreover, by the linearity of the utility function, if a mixed strategy has maximum

expected utility for the defender, then so do all pure strategies in support of that mixed strategy. Thus, a_1^* is the strategy with maximum expected utility, given the probability distribution over attacker types as well as the predicted strategy for each type of attacker. □

Theorem 6 means that a maximally risk-seeking defender just needs to consider how to maximize their expected utility. Thus, the Bayesian Nash Equilibrium in the static Bayesian game is a special case of our acceptable costs of minimax regret, where all players are assumed to be maximally risk-seeking.

By Theorem 6, we have the following theorem to show that our solution concept can give a less conservative strategy than the maximin strategy for the defender in case that the attacker's strategy has been correctly predicted:

Theorem 7. *If the loss tolerance degree for a defender is $\sigma_1 = 1$ and the optimal mixed strategy for the attacker of type t is $\Delta_2^{t,*}$, then the expected utility of the defender's optimal pure strategy a_1^* in our acceptable costs of minimax regret equilibrium is less conservative than the defender's maximin strategy $\underline{\Delta_1}$. That is, in any situation, we have*

$$\sum_{t\in T} \mathsf{p}(t)\mathsf{u}_1^t(a_1^*, \Delta_2^{t,*}) \geq \sum_{t\in T} \mathsf{p}(t)\mathsf{u}_1^t(\underline{\Delta_1}, \Delta_2^{t,*}).$$

And there exists some cases that we have

$$\sum_{t\in T} \mathsf{p}(t)\mathsf{u}_1^t(a_1^*, \Delta_2^{t,*}) > \sum_{t\in T} \mathsf{p}(t)\mathsf{u}_1^t(\underline{\Delta_1}, \Delta_2^{t,*}).$$

Proof. In an SDSRA security game, let $\underline{\Delta_1}$ be the defender's maximin strategy and a_1^* be the defender's optimal pure strategy. Then by Definition 1 and the fact that in SDSRA security games, we only assign utility values based on whether or not an asset is covered. Thus, for any pure strategy a_1 of the defender, there always exists a mixed strategy Δ_1 that assigns a positive value over each pure strategy $a_1 \in A_1$ such that

$$\min_{\Delta_2} \mathsf{u}_1(\Delta_1, \Delta_2) > \min_{\Delta_2} \mathsf{u}_1(a_1, \Delta_2).$$

Thus, by the concept of maximin strategy revealed by formula 6 and the fact that a_1^* is a pure strategy, we have $\underline{\Delta_1} \neq a_1^*$. As a result, by Theorem 6, we have

$$\sum_{t\in T} \mathsf{p}(t)\mathsf{u}_1^t(a_1^*, \Delta_2^{t,*}) \geq \sum_{t\in T} \mathsf{p}(t)\mathsf{u}_1^t(\underline{\Delta_1}, \Delta_2^{t,*}).$$

Finally, by Definition 1 and formula (1), for any SDSRA game with the optimal mixed strategy for the attacker of type t is $\Delta_2^{t,*}$, we can always find a set of utility functions $\{\mathsf{u}_1^t\}$

for defender, such that there exists a pure strategy a_1^*, which satisfies that for any $a_1' \neq a_1^*$, we have

$$\sum_{t \in T} \mathsf{p}(t)\mathsf{u}_1^t(a_1^*, \Delta_2^{t,*}) > \sum_{t \in T} \mathsf{p}(t)\mathsf{u}_1^t(a_1', \Delta_2^{t,*}).$$

As a result, for any mixed strategy $\Delta_1 \neq a_1^*$, we have

$$\sum_{t \in T} \mathsf{p}(t)\mathsf{u}_1^t(a_1^*, \Delta_2^{t,*}) > \sum_{t \in T} \mathsf{p}(t)\mathsf{u}_1^t(\Delta_1, \Delta_2^{t,*}).$$

By the fact that $\underline{\Delta_1} \neq a_1^*$, we have

$$\sum_{t \in T} \mathsf{p}(t)\mathsf{u}_1^t(a_1^*, \Delta_2^{t,*}) \geq \sum_{t \in T} \mathsf{p}(t)\mathsf{u}_1^t(\underline{\Delta_1}, \Delta_2^{t,*}).$$

□

Finally, we have the following theorem which improves the efficiency for counting the defender's optimal strategy with our solution concept by allowing us to only consider the defender's set of pure strategies:

Theorem 8. *The optimal strategy for the defender Δ_1^* in an SDSRA security game is a pure strategy in the defender's set of pure strategies A_1.*

Proof. Suppose $A_1 = \{a_1^1, \ldots, a_1^n\}$ is the defender's set of pure strategies and $A_1' \subset A_1$ is a set of pure strategies which, for any $a_1^i \in A_1'$, satisfies formulas (15) and (16). Then, the mixed strategy which satisfies formulas 15 and 16 is a probability distribution q over $A_1 = \{a_1^1, \ldots, a_1^n\}$, such that for any $a_1^i \in A_1'$, we have $q(a_1^i) \geq 0$ and for any $a_1^j \in A_1 \backslash A_1'$, we have $q(a_1^j) = 0$. Moreover, by formulas (15) and (16), $q(a_1^i) \in [0, 1]$ for any $a_1^i \in A_1'$. Finally, given the linearity of the value of the expected utility functions, by formula (14), we obtain this result directly. □

Theorem 8 says that we only need to consider the defender's set of pure strategies in order to determine their optimal strategy by Definition 4.

6.2. Linear Programming

Given Definitions 3 and 4 as well as Theorems 1, 3 and 8, the whole process of finding a defender's optimal pure strategy can be solved by Linear Programming (*i.e.*, LP 1) and Mixed Integer Linear Programming (*i.e.* MILP 2). In particular, LP 1 describes how to predict the mixed strategy Δ_2^t that will be selected by each type of attacker, while MILP 2 describes how to select the optimal pure strategy a_1 such that $\Delta_1(a_1) = 1$ for the defender based on this prediction. In each program, free indices denote universal quantification over

$$
\begin{aligned}
\min \quad & R^t \\
\text{s.t.} \quad & R^t \geq \max_{a_2' \in A_2} \left(\mathsf{u}_2^t(a_1, a_2')\right) - \sum_{a_2 \in A_2} \Delta_2^t(a_2)\mathsf{u}_2^t(a_1, a_2) && (for\ any\ a_1 \in A_1) \\
& \sum_{a_2 \in A_2} \Delta_2^t(a_2)\mathsf{u}_2^t(a_1, a_2) \geq (1 - \sigma_2^t)B_2^t + \sigma_2^t V_2^t && (for\ any\ a_1 \in A_1) \\
& B_2^t = \sum_{a_2 \in A_2} \epsilon_2^t(a_2)\mathsf{u}_2^t(a_1, a_2) && (for\ any\ a_1 \in A_1) \\
& V_2^t = \min_{a_1 \in A_1} \min_{a_2 \in A_2} \mathsf{u}_2^t(a_1, a_2) \\
& \sum_{a_2 \in A_2} \epsilon_2^t(a_2) = 1 \\
& \epsilon_2^t(a_2) \in [0, 1] \\
& \sum_{a_2 \in A_2} \Delta_2^t(a_2) = 1 \\
& \Delta_2^t(a_2) \in [0, 1]
\end{aligned}
$$

LP 1: Finding attacker type t's optimal mixed strategy Δ_2^t where $\epsilon_2^t(a_2)$ is a mixed strategy representing the equalizer for t.

$$
\begin{aligned}
\max \quad & \sum_{t \in T} \sum_{a_1 \in A_1} \sum_{a_2 \in A_2} \mathsf{p}(t) \Delta_1(a_1) \Delta_2^t(a_2) \mathsf{u}_1^t(a_1, a_2) \\
\text{s.t.} \quad & \sum_{t \in T} \sum_{a_1 \in A_1} \mathsf{p}(t) \Delta_1(a_1) \mathsf{u}_1^t(a_1, a_2) \geq (1 - \sigma_1) B_1 + \sigma_1 V_1 \qquad (for\ any\ a_2 \in A_2) \\
& 0 \leq B_1 - \sum_{t \in T} \mathsf{p}(t) \min_{a_2} \mathsf{u}_1^t(a_1, a_2) \leq M(1 - \Delta_1'(a_1)) \quad (for\ any\ \Delta_1'(a_1)\ over\ A_1) \\
& V_1 \leq \sum_{t \in T} \mathsf{p}(t) \min_{a_2} \mathsf{u}_1^t(a_1, a_2) \qquad (for\ any\ a_1 \in A_1) \\
& \sum_{a_1 \in A_1} \Delta_1(a_1) = 1 \\
& \Delta_1(a_1) \in \{0, 1\} \\
& \sum_{a_1 \in A_1} \Delta_1'(a_1) = 1 \\
& \Delta_1'(a_1) \in \{0, 1\}
\end{aligned}
$$

MILP 2: Finding the defender's optimal pure strategy a_1 such that $\Delta_1(a_1) = 1$ where Δ_2^t is the predicted optimal mixed strategy for attacker type t and M is some large constant.

constraints with the exception of attacker type $t \in T$ in LP 1, (*e.g.*, the first constraint in LP 1 is repeated for each pure strategy $a_1 \in A_1$). We will now briefly summarize these programs.

In LP 1, the objective function and the first constraint represent formula (10) while the second, third and fourth constraints represent formulas (12) and (13), where ϵ_2^t is a mixed strategy representing the equalizer for an attacker of type t, B_2^t is the maximin expected utility for the attacker with type t, and V_2^t is the minimum expected utility for the attacker with type t. The inputs to LP 1 are the utility of each pure strategy profile $\mathsf{u}_2^t(a_1, a_2)$ for each type of attacker, the loss tolerance degree σ_2^t for each type of attacker, and the maximin strategy $\epsilon_2^t(a_2)$. These inputs comprise the definition of an SDSRA problem as in Definition 1, and the loss-aversion attitude of each type of attacker. The output from LP 1 is then a mixed strategy Δ_2^t.

In MILP 2, the objective function represents formula (14), while the first, second and third constraints represent formulas (15) and 16, where B_1 is the maximin expected utility for the defender and V_1 is the minimum expected utility for the defender. Moreover, the fourth constraint limits the strategy selected by the defender to a pure distribution over A_1, (*i.e.*, either $\Delta_1(a_1) = 1$ or $\Delta_1(a_1) = 0$). The inputs for MILP 2 are the optimal mixed strategy Δ_2^t for each type of attacker obtained by LP 1, the probability distribution $\mathsf{p}(t)$ over attacker types, the defender's utility for each pure strategy profile $\mathsf{u}_1^t(a_1, a_2)$, and the defender's loss tolerance degree σ_1. These inputs comprise the definition of an SDSRA problem as in Definition 1, and the loss-aversion attitude of the defender. The output from MILP 2 is then a pure strategy Δ_1.

7. Evaluation

In this section we will illustrate our linear programming model firstly, and then we will prove the robustness of our results and present some experimental results to demonstrate their real-world viability.

7.1. Scenario

Now, suppose in the security game for SDSRA in Examples 1 and 2, the defender has a loss tolerance degree of 0.5 while attacker types t_1, t_2 and t_3 have loss tolerance degrees of 0.8, 0.5 and 0.2, respectively. The game is then solved using LP 1 and MILP 2 in order to determine the defender's optimal pure strategy. Assume a threat has been detected such that $\mathsf{p}(t_1) = 0.2$, $\mathsf{p}(t_2) = 0.3$ and $\mathsf{p}(t_3) = 0.5$.

Now we consider LP 1. The meaning of the objective function is that we want to minimize the value of some variable $R^t \in \mathbb{R}$ for attacker type t. The first constraint imposes a restriction on the possible value of R^t with respect to t's utility values u_2^t and

some other variable Δ_2^t, where the seventh and eighth constraints impose a restriction that Δ_2^t be a probability distribution over the attacker's set of pure strategies A_2, (*i.e.*, that Δ_2^t be a mixed strategy). This first constraint is repeated for each $a_1 \in A_1$. That is, if the defender plays strategy $a_1 \in A_1$, then an instance of this constraint says that the value of R^t must be no less than the difference between t's maximum utility for a_1, given any pure strategy $a_2 \in A_2$, and t's expected utility for a_1, given the mixed strategy Δ_2^t. For example, consider attacker type t_1 from Table 2a such that $t = t_1$. Suppose $a_1 = SM$ and $\Delta_2^t(SM) = \Delta_2^t(FCE) = \Delta_2^t(H) = \frac{1}{3}$, then

$$\max_{a_2 \in A_2} \mathsf{u}_2^t(a_1, a_2) = \max\{-7, 3, 5\} = 5,$$
$$\sum_{a_2 \in A_2} \Delta_2^t(a_2)\mathsf{u}_2^t(a_1, a_2) = \frac{1}{3} \cdot (-7) + \frac{1}{3} \cdot 3 + \frac{1}{3} \cdot 5 = \frac{1}{3}.$$

Thus, their difference is $5 - \frac{1}{3} = \frac{14}{3}$. Similarly, the difference between the maximum utility and $\Delta_2^t(SM) = \Delta_2^t(FCE) = \Delta_2^t(H) = \frac{1}{3}$ for $a_1 = FCE$ is 5 and for $a_1 = H$ is 7. In other words, for these values of Δ_2^t, we have constraints such that $R^t \geq \frac{14}{3}$, $R^t \geq 5$, and $R^t \geq 7$ meaning that R^t must be at least 7. Obviously, if we change the values of Δ_2^t, then the lower boundary of R^t will change. Thus, the aim of LP 1 is to find the values of Δ_2^t which support the lowest boundary of R^t, given the utility values u_2^t.

The second constraint in LP 1 imposes a restriction on the possible values of Δ_2^t such that the expected utility for Δ_2^t (*i.e.*, $\sum_{a_2 \in A_2} \Delta_2^t(a_2)\mathsf{u}_2^t(a_1, a_2)$) must be not lower than some acceptable minimal payoff values $(1 - \sigma_2^l)B_2^l + \sigma_2^l V_2^l$. Again, the second constraint is repeated for each $a_1 \in A_1$. The acceptable minimal payoff itself is defined by the third and fourth constraints. In fact, this acceptable minimal payoff acts as a constant for t based on their loss tolerance degree σ_2^t where B_2^t is the maximin expected utility for t and V_2^t is the minimum utility for t. More specifically, for the maximin expected utility B_2^t, we need to find a mixed strategy Δ_2^t such that their expected utilities are the same, regardless of the strategy selected by the defender. For example, consider again attacker type t_1 from Table 2a such that $t = t_1$. To determine the value of B_2^l, we first need to find values for $\epsilon_2^t(SM)$, $\epsilon_2^t(FCE)$, and $\epsilon_2^t(H)$ such that $\epsilon_2^t(SM) + \epsilon_2^t(FCE) + \epsilon_2^t(H) = 1$ and

$$\begin{aligned} & \epsilon_2^t(SM) \cdot (-7) + \epsilon_2^t(FCE) \cdot (3) + \epsilon_2^t(H) \cdot (5) \\ = \; & \epsilon_2^t(SM) \cdot (9) + \epsilon_2^t(FCE) \cdot (-2) + \epsilon_2^t(H) \cdot (5) \\ = \; & \epsilon_2^t(SM) \cdot (9) + \epsilon_2^t(FCE) \cdot (3) + \epsilon_2^t(H) \cdot (-6). \end{aligned}$$

This is called the equalizer and we have $\epsilon_2^t(SM) = \frac{55}{311}$, $\epsilon_2^t(FCE) = \frac{176}{311}$ and $\epsilon_2^t(H) = \frac{80}{311}$. Thus, $B_2^l = \frac{543}{311}$. For V_2^l, the value can be obtained directly from the utility matrix. In this case, $V_2^l = \min\{-7, -6, -2, 3, 5, 9\} = -7$. Given that $\sigma_2^{t_1} = 0.8$, then the acceptable

minimal payoff for t_1 is $0.2 \cdot \frac{543}{311} + 0.8 \cdot (-7) = -0.21$. Finally, if we model the constraints for t_1 in full, then we arrive at the following LP problem:

$$
\begin{aligned}
\min \quad & R^t \\
\text{s.t.} \quad & R^t \geq 5 - (\Delta_2^t(SM) \cdot (-7) + \Delta_2^t(FCE) \cdot (3) + \Delta_2^t(H) \cdot (5)) \\
& R^t \geq 9 - (\Delta_2^t(SM) \cdot (9) + \Delta_2^t(FCE) \cdot (-2) + \Delta_2^t(H) \cdot (5)) \\
& R^t \geq 9 - (\Delta_2^t(SM) \cdot (9) + \Delta_2^t(FCE) \cdot (3) + \Delta_2^t(H) \cdot (-6)) \\
& \Delta_2^t(SM) \cdot (-7) + \Delta_2^t(FCE) \cdot (3) + \Delta_2^t(H) \cdot (5) \geq -0.21 \\
& \Delta_2^t(SM) \cdot (9) + \Delta_2^t(FCE) \cdot (-2) + \Delta_2^t(H) \cdot (5) \geq -0.21 \\
& \Delta_2^t(SM) \cdot (9) + \Delta_2^t(FCE) \cdot (3) + \Delta_2^t(H) \cdot (-6) \geq -0.21 \\
& \sum_{a_2 \in A_2} \Delta_2^t(a_2) = 1 \\
& \Delta_2^t(a_2) \in [0, 1]
\end{aligned}
$$

By solving the above problem with an LP solver, we will find that $\Delta_2^t(SM) = 0.361$, $\Delta_2^t(FCE) = 0.439$, and $\Delta_2^t(H) = 0.2$. Thus, by the principle of acceptable costs of minimax regret, we predict that t_1 will play this mixed strategy. Similarly, if we apply this process for t_2 and t_3, we will obtain the mixed strategies $\{\Delta_2^{t_2}(SM) = 0.325, \Delta_2^{t_2}(FCE) = 0.35, \Delta_2^{t_2}(H) = 0.325\}$ and $\{\Delta_2^{t_3}(SM) = 0.349, \Delta_2^{t_3}(FCE) = 0.326, \Delta_2^{t_3}(H) = 0.326\}$, respectively.

Now consider MILP 2. The meaning of the objective function is that we want to find a pure strategy which can maximize the defender's expected utility according to the mixed strategy prediction for each attacker type. However, the first constraint again imposes a restriction that the chosen strategy satisfies an acceptable minimal payoff obtained by using the defender's loss tolerance degree which, in this case, is $\sigma_1 = 0.5$. By referring to the defender's utility values, the probability distribution over attacker types and the predicted mixed strategy for each attacker type, we arrive at the following objective function:

$$
\begin{aligned}
\max \quad & E^{t_1} + E^{t_2} + E^{t_3} \\
\text{s.t.} \quad & E^{t_1} = 0.2 \cdot (E_{SM}^{t_1} + E_{FCE}^{t_1} + E_H^{t_1}) \\
& E^{t_2} = 0.3 \cdot (E_{SM}^{t_2} + E_{FCE}^{t_3} + E_H^{t_3}) \\
& E^{t_3} = 0.5 \cdot (E_{SM}^{t_3} + E_{FCE}^{t_3} + E_H^{t_3}) \\
& E_{SM}^{t_1} = \Delta_1(SM) \cdot (0.361 \cdot 7 + 0.439 \cdot (-3) + 0.2 \cdot (-6)) \\
& E_{FCE}^{t_1} = \Delta_1(FCE) \cdot (0.361 \cdot (-8) + 0.439 \cdot 4 + 0.2 \cdot (-6)) \\
& E_H^{t_1} = \Delta_1(H) \cdot (0.361 \cdot (-8) + 0.439 \cdot (-3) + 0.2 \cdot 6) \\
& E_{SM}^{t_2} = \Delta_1(SM) \cdot (0.325 \cdot 5 + 0.35 \cdot (-6) + 0.325 \cdot (-5))
\end{aligned}
$$

$$E^{t_2}_{FCE} = \Delta_1(FCE) \cdot (0.325 \cdot (-4) + 0.35 \cdot 7 + 0.325 \cdot (-5))$$
$$E^{t_2}_{H} = \Delta_1(H) \cdot (0.325 \cdot (-4) + 0.35 \cdot (-6) + 0.325 \cdot 6)$$
$$E^{t_3}_{SM} = \Delta_1(SM) \cdot (0.349 \cdot 5 + 0.326 \cdot (-4) + 0.326 \cdot (-6))$$
$$E^{t_3}_{SM} = \Delta_1(FCE) \cdot (0.349 \cdot (-5) + 0.326 \cdot 5 + 0.326 \cdot (-6))$$
$$E^{t_3}_{SM} = \Delta_1(H) \cdot (0.349 \cdot (-5) + 0.326 \cdot (-4) + 0.326 \cdot 7)$$

The second constraint then imposes comparable restrictions on the possible pure strategy which can be selected to those described for LP 1. Once we have imposed the constraints and solved the problem with an LP solver, we will find that $\Delta_1(SM) = 1$ and $\Delta_1(FCE) = \Delta_1(H) = 0$. Thus, according to the principle of acceptable costs of minimax regret, the defender's optimal pure strategy for this game is SM.

7.2. Analyzing Robustness

In a linear program, it is often the case that some (or all) of the right-hand sides of the constraints are subject to sources of uncertainty, including errors of measurement, absence of information, and so on. This uncertainty will therefore reduce our confidence in the solution of the linear programming. Good modeling practice requires an evaluation of how optimum solutions change if we modify the constraints and the robustness of the linear programming in the presence of uncertainty. In this subsection, we will carry out such an evaluation with respect to our solution to SDSRA problems.

In SDSRA problems, the loss tolerance degree for each player is an absolute value and is based on expert's subjective judgements and historical data. Therefore, SDSRA problems are clearly subject to uncertainty. It is useful to consider how small changes to a loss tolerance degree might change the strategy selected by the attacker, as well as the robustness of the defender's expected utility for their optimal strategy. Firstly, we want to show that a small change to the loss tolerance degree of the attacker does not cause a dramatic change to the strategy selected by the attacker. In other words, we want to show that the solution by Definition 3 is continuous when the loss tolerance degree of the attacker is in the range $[0, 1]$. In order to prove this, we need to consider the existence of the optimal solution for each type of attacker based on Definition 3. By Theorem 1, we have the following:

Lemma 1. *In an SDSRA security game, each type of attacker can always select a strategy* $\underline{\Delta_2}$ *as a feasible strategy which satisfies:*

$$\min_{a_1} \mathsf{u}^t_2(a_1, \underline{\Delta_2}) \geq \left(\max_{\Delta_2} \min_{a_1} \mathsf{u}^t_2(a_1, \Delta_2) \right) - \varsigma_2,$$

$$\varsigma_2 = \sigma_2^t \left(\max_{\Delta_2} \min_{a_1} \mathsf{u}_2^t(a_1, \Delta_2) - \min_{\Delta_2'} \min_{a_1'} \mathsf{u}_2^t(a_1', \Delta_2') \right),$$

where Δ_2' and Δ_2'' are mixed strategies for the attacker and $\sigma_2^t \in [0,1]$ is a loss tolerance degree.

Moreover, the existence of the optimal solution for each attacker type can be proved by the following:

Theorem 9. *In an SDSRA security game, each type of attacker can always find an optimal strategy that satisfies Definition 3 by LP 1.*

Proof. Suppose $\sigma_2^t \in [0,1]$ is a degree of loss tolerance for an attacker of type t, $v(\gamma(\Delta_a^t))$ is the maximum regret value of the minimax regret strategy $\gamma(\Delta_a^t)$ of attacker of type t, and $v(\underline{\Delta_a^t})$ is the maximum regret value of the maximin strategy $\underline{\Delta_a^t}$ of attacker of type t. Then, by Lemma 1, $\underline{\Delta_a^t}$ is a feasible solution to LP 1. Thus, when $\sigma_2^t \in [0,1]$, the feasible solution set of LP 1 exists. Finally, if the feasible solution set of LP 1 exists, that the probability value of each pure strategy of a given type of attacker is in the interval $[0,1]$, and the optimal objective value of LP 1 is in the interval $[v(\underline{\Delta_a^t}), v(\gamma(\Delta_a^t))]$, then given the linearity of expected utility functions with the probability distribution over the attacker types, we can find that the optimal solution of each type of attacker exists straightforward. □

In order to make the proof of the next theorem easier to understand, we introduce a translation of a normal linear programming problem into an *augmented form* (slack form) parametric linear programming problem. Here an augmented form means to replace inequalities with equalities in the constraints by introducing non-negative slack variables. [6] An example of the augmented form translation is given as follows:

$$\begin{array}{lll} \max & z = -x_1 + x_2 & \\ \text{s.t.} & 2x_1 - x_2 \geq -2 & \\ & x_1 - 2x_2 \leq 2 & \Rightarrow \\ & x_1 + x_2 \leq 5 & \\ & x_1 \geq 0 & \end{array} \qquad \begin{array}{lll} \min & -z = x_1 - (x_3 - x_4) & \\ \text{s.t.} & 2x_1 - (x_3 - x_4) - x_5 = -2 & \\ & x_1 - 2(x_3 - x_4) + x_6 = 2 & \\ & x_1 + (x_3 - x_4) + x_7 = 5 & \\ & x_i \geq 0, i = 1, \ldots, 7 & \end{array}$$

Based on such translation, we have the following theorem, which means the solution for LP 1 is continuous:

[6] We refer the reader to (Schrijver, 1998) for more details about this translation.

Theorem 10. *Let $\sigma_2^t \in [0,1]$ be the loss tolerance degree for the attacker of type t and $\Delta_2^{t,*}$ be the optimal mixed strategy for t. Then $\Delta_2^{t,*}$ is a continuous vector function or a continuous point-to-set mapping.*

Proof. First, we will translate our linear programming problem in LP 1 into an *augmented form* (slack form) parametric linear programming problem based on the loss tolerance degree σ_2^t. Then by LP 1, suppose $\Delta_2^t = \{x_1, \ldots, x_n\}$ is a mixed strategy for attacker type t,

$$k_j = \max_{a_2 \in A_2} \mathsf{u}_2^t(a_1^j, a_2)$$

for each pure strategy a_1^j of the defender such that $j = 1, \ldots, n$, $x_{3n+1} - x_{3n+2}$ is the maximum regret value R^t found in LP 1, $x_{n+1}, \ldots, x_{3n}$ are surplus variables required to translate the inequalities into equalities, and B_2^t and V_2^t are the same constants as those in LP 1. Then we can obtain a parametric linear programming based on the loss tolerance degree σ_2^t which takes the augmented form as follows:

$$\begin{aligned}
\min \quad & x_{3n+1} - x_{3n+2} \\
\text{s.t.} \quad & x_{3n+1} - x_{3n+2} + \sum_{x_i=1}^{n} x_i \mathsf{u}_2^t(a_1, a_2') - x_{n+j} = k_j + \sigma_2^t \times 0 \\
& \sum_{x_i=1}^{n} x_i = 1 + \sigma_2^t \times 0 \\
& \sum_{x_i=1}^{n} x_i \mathsf{u}_2^t(a_1, a_2') - x_{2n+l} = B_2^t + \sigma_2^t (V_2^t - B_2^t) \\
& x_m \geq 0
\end{aligned}$$

where $l = 1, \ldots, n$ and $m = 1, \ldots, 3n+1, 3n+2$.

Then, we can find that such augmented form satisfies the following form:

$$\begin{aligned}
\min \quad & cx \\
\text{s.t.} \quad & Ax = b_1 \sigma_2^t + b_2 = b(\sigma_2^t) \\
& x \geq 0 \\
& \sigma_2^t \in \mathbb{R} \\
& b_1, b_2 \in \mathbb{R}^m \\
& c, x \in \mathbb{R}^n
\end{aligned}$$

where A is an $m \times n$ matrix with such that $\mathsf{rank}(A) = m$. Moreover, by Theorem 9, LP 1 can always find the optimal solution when $\sigma_2^t \in [0,1]$. Thus, depending on degeneracy

of the parametric linear programming based on the loss tolerance degree (*i.e.*, LP 1), the solution to LP 1 is a continuous vector function or a continuous point-to-set mapping based on σ_2^t according to the proof in (Zhang and Liu, 1990). □

Actually, Theorem 10 demonstrates the robustness of our solution concept when selecting the attacker's strategy, in the sense that a small change of their loss tolerance degree does not make a big difference on the result.

We can now consider the robustness of our solution concept when selecting the defender's optimal strategy. There are three aspects: (i) robustness with respect to the worst case; (ii) robustness with respect to uncertainty over the loss tolerance degree for each type of attacker; and (iii) robustness with respect to a deviation from the loss tolerance degree for each type of attacker. Let us first consider aspect (i). By Definition 4 in our solution concept, we have that formulas (15) and (16) guarantee that the minimum expected utility for a given pure strategy is acceptable for the defender. That is, even if the defender plays against an arbitrary attacker, they can always select a maximin strategy to guarantee their minimum expected utility when the attacker is playing the worst-case strategy. Therefore, our solution concept is robust regarding aspect (i). Let us now consider aspects (ii) and (iii). Since the minimum expected utility of the minimax regret strategy for the attacker is always higher than the minimin expected utility of the attacker, we can determine a range by sensitivity analysis of the linear program (Schrijver, 1998) over which the loss tolerance degree for a given type attacker will not change the optimal mixed strategy. Formally, we have

Corollary 1. *Let $\gamma(\Delta_a^t)$ be the minimax regret strategy for an attacker of type t, σ_2^t be the loss tolerance degree for t, $\overline{u_a^t}$ be the maximin expected utility for t, $\underline{u_a^t}$ be the minimin expected utility for t, and $\Delta_2^{t,*}$ be the optimal mixed strategy for t. Then $\Delta_2^{t,*}$ remains the same if $x \leq \sigma_2^t \leq 1$, where*

$$x = \frac{\overline{u_a^t} - \min\limits_{a_1} \mathsf{u}_2^t(a_1, \gamma(\Delta_a^t))}{\overline{u_a^t} - \underline{u_a^t}}.$$

If the defender's estimation of the loss tolerance degree of a given type of attacker is in the interval outlined in Corollary 1 (*i.e.*, $\sigma_2^t \in [x, 1]$) then the defender's optimal strategy does not need to change. Therefore, Corollary 1 reveals the degree of uncertainty or deviation over the attacker's loss tolerance degree that the defender can tolerate. In other words, Corollary 1 reflects both aspects (ii) and (iii) in our robustness analysis.

7.3. Methods Comparison

In this subsection, we will give a summary of the comparison of our method with different methods in the security game for SDSRA.

Generally speaking, although there are various methods for different applications in the security game, we can still distinguish them according to the operation mechanisms behind them as follows: (i) Stackelberg Game Framework (SGF): the attacker has at least partial knowledge of the defender's strategy commitment and the attacker will act after they learn the selection of the defender, such as the methods in (Tambe, 2011; Yang et al., 2012; Jiang et al., 2013; Nguyen et al., 2013; Fang et al., 2015; Sinha et al., 2016), etc.. (ii) Nash Equilibrium Framework (NEF): attacker and defender both have complete knowledge about the strategy selected by each other and both players act simultaneous, such as the method in (Korzhyk et al., 2011). (iii) Machine Learning Framework (MLF): it requires significant amounts of historical data about players interactions in order to learn a reasonably representative model of adversary behavior. Then, the defender selects the strategy with maximum expected utility as their optimal strategy. (iv) Quantal Response Framework (QRF): in this framework, attackers are assumed to make errors in choosing which pure strategy to play. The probability of any particular strategy being chosen is positively related to the payoff from that strategy, such as the method in (Yang et al., 2012). (v) Worst Case Framework (WCF): the attacker may act randomly and the defender just wants to guarantee their minimum utility in the worst case, such as the methods proposed by Kiekintveld and Kreinovich (2012), Nguyen et al. (2014a), and Pita et al. (2010). In this framework, the defender selects their strategy based on the maximin strategy.

In order to show the advantages of our method in the security game of SDSRA in Definition 1, we examine the following aspects: (i) Knowledge (of the defender's strategy commitment): since in SDSRA problems, the assumption of complete or partial knowledge of the defender's strategy commitment is unrealistic as argued in Section 2, it is important to judge whether a mechanism can be used to solve the SDSRA problems or not. (ii) Training Data: since there is very limited data available in infrastructure security domains (which we focus on in SDSRA problems), it is impossible to obtain enough training data to analyse the behavior of the attacker in SDSRA problem. (iii) Prediction Power: since the defender needs to find out their optimal strategy based on the prediction of the attackers' strategies in SDSRA problems, the prediction power of the attackers' behavior determines the quality of the defender's action in threat prevention. (iv) Robust: since the attacker's strategy is based on a judgement of the attacker's utility matrix, the loss tolerance assumption for each type of attacker, and imperfect information obtained by surveillance system, there is a chance that the attacker may play a strategy different from the strategy predicted by the defender. Therefore, we need a mechanisms that can properly deal with this sort of uncertainty and avoid unacceptable losses in the worst case. (iv) Solution Quality: if two mechanisms can both be used to solve SDSRA problems, we need to consider which one will give the defender a higher expect utility. Now, based on such consideration, we can compare our method with these four types of operation mechanisms

	Knowledge	Training Data	Prediction Power	Robust	Solution Quality
SGF	Complete or partial $(\times)$	Unnecessary	Weak$(\times)$	Weak$(\times)$	-
NEF	Complete $(\times)$	Unnecessary	Weak$(\times)$	Weak$(\times)$	-
MLF	Unnecessary	Required $(\times)$	Weak$(\times)$	Weak$(\times)$	-
QRF	Unnecessary	Unnecessary	Weak $(\times)$	Weak $(\times)$	-
WCF	Unnecessary	Unnecessary	Unnecessary	Strong	Lower$(\times)$
Our's	Unnecessary	Unnecessary	Strong	Strong	Higher

Table 6: Methods Comparasion

as shown in Table 6.

In Table 6, the symbol of $(\times)$ means that a given operation mechanism is unsuitable for solving the SDSRA problems or there exists a mechanism better than the given mechanism in an aspect. And the symbol of $-$ means that since the mechanism is unsuitable for solving the SDSRA problems or there exists a mechanism better than the given mechanism in some aspects, we do not consider its solution quality. Moreover, for the second row of Table 6, we can find that since Stackelberg game framework requires complete or partial knowledge of the defender's strategy commitment, its prediction power for the SDSRA problems is weak. Also, since the Stackelberg game framework only focuses the maximum expected utility of the defender, its robust is weak. From the third row of Table 6, similarly to SGF, we can find that Nash Equilibrium framework requires complete knowledge of the defender's strategy commitment, its prediction power for the SDSRA problems is weak, and its robust is weak. In the fourth row, we consider the machine learning framework. For this framework, since it requires a large training data to analyse the behavior of the attackers but there is very limited data available in SDSRA problems, its prediction power and robust is weak. In the fifth row, we analyse the quantal response framework. Although this framework does not require the knowledge of the defender's strategy commitment or any training data, its assumption that the attacker's ignorance over the defender's strategy should be modeled by the principle of indifference is not on the line of our intuition and the human behaviors in game theory revealed by psychological experiments and analysis of economics (Ellsberg, 1961; Kahneman and Tversky, 1979; Kahneman, 2003; Savage, 1951). Actually, the behavior of the column player in the Aumann and Maschler Game in Table 3 and that in the Asymmetric matching pennies game in Table 4 indeed show that in some situations, it is unrealistic to assume that the row player's ignorance about the column player should be modeled by a mixed strategy which assigns equal probabilities to all pure strategies. Thus, we doubt the prediction power of the quantal response framework can properly solve the SDSRA problem. In the sixth row, we find that although the worst case framework does not require the knowledge of the defender's strategy commitment or

any training data and its performance in prediction power and robust is acceptable for the SDSRA problem, by Theorem 7, it is conservative than our solution concept. Thus, we do not think such framework is suitable for solving the SDSRA problem since our model can always get an expected payoff for the defender not less than this framework.

Finally, for the seventh row, we can easily find that our solution concept satisfies the limitations of the SDSRA problem is the aspects of knowledge about the defender's strategy commitment and training data by the rationalizability of our solution concept in the SDSRA problem that revealed by Section 5. And by the rationalizability of our solution concept in Section 5, Theorems 2 and 5, and the fact about human behaviors in game theory revealed by psychological experiments and analysis of economics (Ellsberg, 1961; Kahneman and Tversky, 1979; Kahneman, 2003; Savage, 1951), we can find that the performance of our method in the aspect of prediction power is better than other operation mechanisms. Hence, by Definition 4 and Section 7.2, we can obtain that the robustness of our method is as strong as the worse case framework. Also, by Theorem 7, we indeed find the performance of our method is better than the worse case framework in the aspect of solution quality. And all of them are the reasons why our solution concept should be used to solve the security game of SDSRA problems.

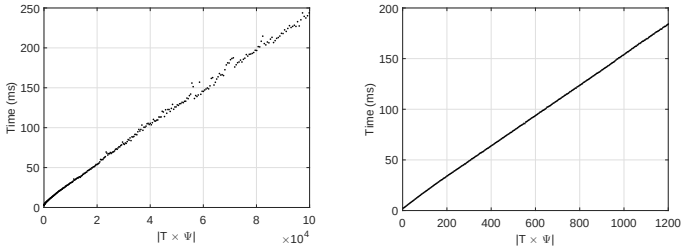
7.4. Experiments

An implementation of our solution concept, called SDSRA, was developed in Java for the IBM ILOG CPLEX OPTIMIZATION STUDIO[7] software and is available online.[8] In order to experimentally evaluate the viability of our solution concept, we have opted for randomly generated SDSRA security games as our data set. All utility values are random integers in the range $[-10, 0]$ or $[0, 10]$, depending on whether a target is covered or uncovered. All experiments were carried out on a desktop computer with an Intel Xeon E5-2620 2GHz CPU and 16GB of RAM running Microsoft Windows 7 Enterprise 64-bit. The software versions used during these experiments were SDSRA 1.0 and IBM ILOG CPLEX OPTIMIZATION STUDIO 12.6.

By Definition 1, we know that the size of the input is based on the number of utility values, which are determined by the set of attacker types T and the set of pure strategy profiles Ψ. Therefore, we will focus on the value of $|T \times \Psi|$ in our experiments (*i.e.*, for LP 1 and MILP 2). Figure 1a shows the results for an experiment on randomly generated SDSRA security games with one type of attacker, where the number of targets is incremented from 2. Figure 1b then shows the results for an experiment on randomly generated SDSRA security games with two targets, where the number of attacker types is increment-

[7] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer

[8] https://github.com/kevinmcareavey/sdsra



(a) One attacker type, *i.e.*, $|T| = 1$.

(b) Two targets, *i.e.*, $|\Psi| = 4$.

Figure 1: Mean time (from 1000 samples) required to find the acceptable costs of minimax regret equilibrium in randomly generated SDSRA security games using LP 1 and MILP 2), where $|T \times \Psi|$ is the total number of utility values for each player.

ed from 1. In each case, we can see that the total time required to determine the defender's optimal pure strategy is linearly increasing with the total number of utilities values for each player, *i.e.*, the product of the set of attacker types T and the set of pure strategy profiles Ψ. Obviously, if there are n targets, then $|\Psi| = n^2$. This is the reason why our data points become more sparse in Figure 1a as the number of targets is increased. However, the results for Figure 1b clearly demonstrate that the number of attacker types has a more significant impact on run time than the number of targets. This result is unsurprising given that we must solve LP 1 for each type of attacker in order to predict their optimal mixed strategy. Nonetheless, these results convincingly evidence that our programs scale to very large games, *e.g.*, Figure 1a shows that a game with 1 attacker type and 100,000 pure strategy profiles can be solved within 250 ms, while Figure 1b shows that a game with 4 pure strategy profiles and 300 attacker types can be solved within 200 ms. Thus, it is very likely that our solution concept is viable for many real-world SDSRA environments.

8. Conclusion

It is very important to support online security resource allocation in intelligence surveillance systems in real-world applications. However, existing work on game-theoretic security resource allocation and its applications has addressed little the subject of surveillance-driven security resource allocation (SDSRA). To this end, in this paper we proposed a new solution concept, called *acceptable costs of minimax regret equilibrium*, for handling game-theoretic SDSRA based on the principle of acceptable costs of minimax regret. The main ideas behind our work are the concepts of loss-aversion and regret. In general, human attackers have different objectives and loss-aversion attitudes. When planning an attack, these play a major role in their decision-making process and thus the strategy to which

they commit. We believe that this critical issue has not been adequately addressed in the literature, and thus in this paper we have explored how these issues affect the prediction of the attacker's strategies as well as the defender's optimal strategy selection. Moreover, we theoretically reveal some essential properties of this solution concept, and formulated the solution concept as an efficient, solvable linear programming problem. The experimental evaluation, prove the robustness of our solution concept, show the advantages of our model for solving SDSRA problem, and based on an extensive implementation of the new solution concept and other related components, provides reassurance that even for very large games, our solution concept remains viable. We believe that the findings in this paper provide some valuable insight into how real-time surveillance information can be used for dynamic security resource allocation. In particular, how regret and different levels of loss-aversion may influence the prediction of an attacker's strategy and the subsequent selection of a strategy for the defender.

There are numerous possibilities for further work. Perhaps the most interesting one is to demonstrate, in an online surveillance environment, that selecting the defender's optimal strategy using our solution concept is preferable to a subjective security team assessment. Thus, an important challenge is to conduct an experimental study of the psychological aspects of our solution concept based on human trials. Another avenue for future work relates to the SDSRA problems with dependent multiple attackers. That is, the attackers in the SDSRA problem can negotiate with each other and try to cooperate or share the profit obtained in an attack. For example, in some terrorist attacks, the attackers work as a team to attack multiple targets and they share the total profit obtained in a series of attacks. Clearly, in this case, the non-cooperative static (simultaneous move) game we constructed in this paper is unsuitable Thus, it is necessary to construct a cooperative game and find out a new solution concept. And in this paper, we have mentioned that the loss tolerance degree of each type of attacker can be learned from data or can be estimated by domain experts and the security manager can fine-tune loss tolerance degree of the defender to reflect different real-time applications. However, we still not give the formal methods to obtain the two values. Thus, it is worth considering some proper methods to estimate the two values. Finally, since the SDSRA problems is based on the surveillance-driven probabilistic information obtained by an intelligent surveillance system, it is interesting to study how to integrate our method with an intelligence surveillance system in practice. Therefore, we plan to extend the event reasoning framework developed in the CSIT project Ma et al. (2010, 2009); Hong et al. (2016) with our method as a means of evaluating our work in practice.

Acknowledgements

This work has been partially funded by EPSRC PACES project (Ref: EP/J012149/1), Foundation for Young Talents in Higher Education of Guangdong (No. 2016KQNCX030), the China Postdoctoral Science Foundation (No. 2017M612688), and South China Normal University Young Teacher Research and Cultivation Fund Project (No. 16KJ13). Also, it was partially undertaken by some authors at Queen's University Belfast, Northern Ireland.

References

Balcan, M.-F., Blum, A., Haghtalab, N., and Procaccia, A. D. (2015). Commitment without regrets: Online learning in Stackelberg security games. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 61–78. ACM.

Blum, A., Haghtalab, N., and Procaccia, A. D. (2014). Learning optimal commitment to overcome insecurity. In *Advances in Neural Information Processing Systems*, pages 1826–1834.

Chua, H. F., Gonzalez, R., Taylor, S. F., Welsh, R. C., and Liberzon, I. (2009). Decision-related loss: Regret and disappointment. *NeuroImage*, 47(4):2031–2040.

Clempner, J. B. (2017). A continuous-time markov stackelberg security game approach for reasoning about real patrol strategies. *International Journal of Control*, pages 1–29.

Crawford, V. P., Costa-Gomes, M. A., and Iriberri, N. (2013). Structural models of nonequilibrium strategic thinking: Theory, evidence, and applications. *Journal of Economic Literature*, 51(1):5–62.

Du, D., Wen, L., Qi, H., Huang, Q., Tian, Q., and Lyu, S. (2018). Iterative graph seeking for object tracking. *IEEE Transactions on Image Processing*, 27(4):1809–1821.

Ellsberg, D. (1961). Risk, Ambiguity, and the Savage Axioms. *Quarterly Journal of Economics*, 75:643–669.

Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. Y. (2004). *Reasoning About Knowledge*. MIT Press.

Fang, F., Stone, P., and Tambe, M. (2015). When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2589–2595.

Goeree, J. K. and Holt, C. A. (2001). Ten little treasures of game theory and ten intuitive contradictions. *American Economic Review*, 91(5):1402–1422.

Goeree, J. K., Holt, C. A., and Palfrey, T. R. (2001). Risk averse behavior in asymmetric matching pennies games. *Games & Economic Behavior*, 45(1):97–113.

Halpern, J. Y. and Pass, R. (2012). Iterated regret minimization: A new solution concept. *Games and Economic Behavior*, 74(1):184–207.

Harsanyi, J. C. (1977). *Rational behavior and bargaining equilibrium in games and social situations*. Cambridge University Press.

Hong, X., Huang, Y., Ma, W., Varadarajan, S., Miller, P., Liu, W., Romero, M. J. S., Rincon, J. M. D., and Zhou, H. (2016). Evidential event inference in transport video surveillance. *Computer Vision & Image Understanding*, 144(C):276–297.

Jaffray, J.-Y. and Jeleva, M. (2007). Information Processing under Imprecise Risk with the Hurwicz criterion. In *Proceedings of the 5th International Symposium on Imprecise Probability: Theories and Applications*, pages 233–242.

Jiang, A. X., Nguyen, T. H., Tambe, M., and Procaccia, A. D. (2013). Monotonic maximin: A robust Stackelberg solution against boundedly rational followers. In *Decision and Game Theory for Security*, pages 119–139. Springer.

Kahneman, D. (2003). A perspective on judgment and choice: Mapping bounded rationality. *American Psychologist*, 58(9):697–720.

Kahneman, D. and Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292.

Kar, D., Fang, F., Delle Fave, F., Sintov, N., and Tambe, M. (2015). A game of thrones: When human behavior models compete in repeated Stackelberg security games. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent System (AAMAS)*, pages 1381–1390.

Kar, D., Fang, F., Fave, F. M. D., Sintov, N., Tambe, M., and Lyet, A. (2016). Comparing human behavior models in repeated stackelberg security games: An extended study . *Artificial Intelligence*, 240:65–103.

Kiekintveld, C., Islam, T., and Kreinovich, V. (2013). Security games with interval uncertainty. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent System (AAMAS)*, pages 231–238.

Kiekintveld, C. and Kreinovich, V. (2012). Efficient approximation for security games with interval uncertainty. In *Proceedings of the AAAI Spring Symposium on Game Theory for Security, Sustainability and Health*, pages 231–238.

Korzhyk, D., Yin, Z., Kiekintveld, C., Conitzer, V., and Tambe, M. (2011). Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *Journal of Artificial Intelligence Research*, 41(2):297–327.

Kuhnen, C. M. and Knutson, B. (2005). The neural basis of financial risk taking. *Neuron*, 47(5):763–770.

Lou, J., Smith, A. M., and Vorobeychik, Y. (2017). Multidefender security games. *IEEE Intelligent Systems*, 32(1):50–60.

Ma, J., Liu, W., and Miller, P. (2010). Event modelling and reasoning with uncertain information for distributed sensor networks. In *Proceedings of the 4th International Conference on Scalable Uncertainty Management*, pages 236–249.

Ma, J., Liu, W., Miller, P., and Yan, W. (2009). Event Composition with Imperfect Information for Bus Surveillance. In *Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 382–387.

Ma, W., Liu, W., Ma, J., and Miller, P. (2014). An extended event reasoning framework for decision support under uncertainty. *Communications in Computer & Information Science*, 444:335–344.

Ma, W., Liu, W., and McAreavey, K. (2015). Game-theoretic resource allocation with real-time probabilistic surveillance information. In *Proceedings of the 13th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 151–161.

Ma, W., Luo, X., and Jiang, Y. (2017). An ambiguity aversion model for decision making under ambiguity. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 614–621.

Ma, W., Luo, X., and Liu, W. (2013a). An ambiguity aversion framework of security games under ambiguities. pages 271–278.

Ma, W., Luo, X., and Liu, W. (2013b). An ambiguity aversion framework of security games under ambiguities. In *Proceedings of the 23d International Joint Conference on Artificial Intelligence*, pages 271–278.

Nguyen, T. H., Jiang, A. X., and Tambe, M. (2014a). Stop the compartmentalization: Unified robust algorithms for handling uncertainties in security games. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent System (AAMAS)*, pages 317–324.

Nguyen, T. H., Yadav, A., An, B., Tambe, M., and Boutilier, C. (2014b). Regret-based optimization and preference elicitation for Stackelberg security games with uncertainty. In *Proceedings of the 28th AAAI*, pages 756–762.

Nguyen, T. H., Yang, R., Azaria, A., Kraus, S., and Tambe, M. (2013). Analyzing the effectiveness of adversary modeling in security games. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 718–724.

Osborne, M. J. (2003). *An Introduction to Game Theory*. Oxford University Press.

Pita, J., Jain, M., Ordóñez, F., Portway, C., Tambe, M., and Western, C. (2009). Using Game Theory for Los Angeles Airport Security. *AI Magazine*, pages 43–57.

Pita, J., Jain, M., Tambe, M., Ordóñez, F., and Kraus, S. (2010). Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence*, 174(15):1142–1171.

Pita, J., John, R., Maheswaran, R., Tambe, M., Yang, R., and Kraus, S. (2012). A robust approach to addressing human adversaries in security games. *Frontiers in Artificial Intelligence & Applications*, 242(2):1297–1298.

Pruzhansky, V. (2011). Some interesting properties of maximin strategies. *International Journal of Game Theory*, 40(2):351–365.

Savage, L. J. (1951). The theory of statistical decision. *Journal of the American Statistical Association*, 46(253):55–67.

Schrijver, A. (1998). *Theory of linear and integer programming*. Wiley.

Sinha, A., Kar, D., and Tambe, M. (2016). Learning adversary behavior in security games: A pac model perspective. In *Proceedings of 15th International Conference on Autonomous Agents and Multiagent System (AAMAS)*, pages 214–222.

Tambe, M. (2011). *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press.

Wasserkrug, S., Gal, A., and Etzion, O. (2008). Inference of security hazards from event composition based on incomplete or uncertain information. *IEEE Transactions on Knowledge and Data Engineering*, 20(8):1111–1114.

Yang, R., Ordóñez, F., and Tambe, M. (2012). Computing optimal strategy against quantal response in security games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent System (AAMAS)*, pages 847–854.

Zhang, X. S. and Liu, D. G. (1990). A note on the continuity of solutions of parametric linear programs. *Mathematical Programming*, 47(1-3):143–153.

Zhang, Y., Zhang, Y., Zhang, Y., Zhang, Y., Zhang, Y., and Zhang, Y. (2016). Game-theory-based active defense for intrusion detection in cyber-physical embedded systems. *Acm Transactions on Embedded Computing Systems*, 16(1):1–21.